ACL Direction Choices

Issues we Need to Resolve Soon.

David Noveck

NFSv4 WG meeting at IETF 120

July 26, 2024

Overview

- Big Changes to ACL description (**Sides 3-5**)
  - Made necessary by state of existing description
  - Entirely new description while retaining comptibility with existing implementations.
  - Includes small protocol extensions as described in Section 9 of RFC8178.
- Need for Consensus (**Slide 6**)
- Consensus Items to Resolve ASAP (**Slides 7-15**)

Big Changes to ACL Description
Slide One of Three

- As part of NFSv4.1 respecification, needed a reasonable way to describe ACLs, including their semantics.
- Simple clarification of the existing description was not possible because:
  - An absence of semantic description had to be filled in.
  - Had to accommodate a number of completely different ACL models
  - Many of the extensions that were added to the core UNIX ACL functionality could not be used.
    - No way for clients to know what extensions were supported by server.

Big Changes to ACL Description
Slide Two of Three

- Major conceptual restructuring required:
  - Explicit specification of core functionality
  - Clarification of semantics:
    - Separation of required semantics and allowed behavioral variants.
  - Extensions to core are considered **OPTIONAL** extensions
- Restructuring allowed/required:
  - Definition of AclChoice attribute to report allowed server choices to client.
    - Allowed **OPTIONAL** extensions to be *used* by clients that needed them, as opposed to them remaining unusable by clients, as currently.

Big Changes to ACL Description
Slide Three of Three

- Restructuring allowed/required (continued):
  - More care in definition of allowed behavioral variants.
  - Need to advise server implementors of choice consequences.
- Performed triage on various ACL models.
  - No time to address them all as part of respecification effort.
  - Core UNIX ACLs need to be more clearly specified
  - Draft POSIX ACLs needs better client support.
  - Extensions for the NFSv4 ACL model will be addressed later.
- Now is the time for the WG to take on the role of triageur.

Need for Consensus

- As decribed previously,
  - We needed to make big changes to previous specifications .
  - Those descriptions were approved as Proposed Standards.
    - Actually, *five times*.
    - We have some hunches, but we don’t why and may never know why.
- We can make such big changes if we have consensus.
- Consensus Items are listed in Appendix A.1.
- Lists of important consensus items in Appendix A.3.
  - Items in Table 9 need to resolved soon on the list.
  - Items in Table 10 concern direction of this effort and need to be resolved ASAP.

Consensus Items to Resolve
#64 -- Role of the Core UNIX ACL Model

- Had to explicitly designate some things as a required part of all implementations.
  - That had been implied but it needed to be an explicit basis for the description.
- Included the following items:
  - Three ACE mask bits that map to the three POSIX privilege bits.
  - ALLOW ACE support required -- support for others **OPTIONAL**
    - Replacing pointless **SHOULD**s in existing specs.
- Those elements are shared with draft POSIX ACLs but more is needed to support those.

Consensus Items to Resolve
#104 – Orphan ACE Mask Bits

- ACE mask bits can be categorized as follows:
  1. The mask bits that correspond to the three POSIX mask bits.
  2. Those that are finer-grained variants of the write bit
  3. Those that are finer grained variants of file ownership
  4. The remainder (“Orphan ACE Mask Bits”)
- Existing specs written assuming #3 and #4 do not exist
- Limited choices for dealing with them.
- Eventually need information about implementations.
- For now, focusing on defining the possible variations and reporting server choices.

Consensus Items to Resolve
#104 – UNIX ACLs as Basis of Description

- Connected with #64 but distinct.
- Current specs treat the full NFSv4 ACL model as canonical.
  - As a result , servers that support anything else are accommodated as an low-quality implementation, of this bigger model, which might not have any full implementations
- Need to restructure the description with UNIX ACLs as the core.
  - Anything beyond that need to be described as an **OPTIONAL** extension, which is what it is.

Consensus Items to Resolve
#105 – Aclchoice Attribute

- Connected with #104 but distinct.
- New **OPTIONAL** attribute implemented as provided for in Section 9 of RFC8178.
- Describes the extensions supported by each server fs.
  - Without that, extensions in the NFSv4 model are not usable.

Consensus Items to Resolve
#109 – RENAME Clarification

- Problems in handling of RENAME authorization.
  - Works OK with coarse-grained authorization (core UNIX ACL model).
- Existing specification assume RENAMEs, require authorization for delete and insert
  - Does not make sense for intra-directory RENAME
  - Hard to imagine user who wants to foreclose RENAME iff he does not allow new files or directories to be created.
- Now a valid behavioral variant.
  - May adjust later based on existing implementations

Consensus Items to Resolve
#110 – Explicit ACL Semantics

- Previous specifications were extremely lax in this regard
  - Part of this was unfamiliarity since previous versions of NFS had left all semantic description to POSIX.
- This became non-viable in V4 because of additions of ACLs and other non-POSIX semantic elements.
  - Left us with a mess given the possibility of behavioral variation allowed by previous specifications.
  - Might nor be possible to unify given existing implementations
  - The most we can do for now is to allow exiting behavioral variation but require servers to report their choices (e.g. via Aclchoice)

Consensus Items to Resolve
#113 – Add Detail about Wgran ACE Mask Bits

- These mask bits fit the varying-granularity model of the existing specs, but …
  - The specs don’t clearly describe how clients can deal with such varying granularity on the server.
- It is possible to describe how clients are to deal with the case in which the client uses a coarse-granularity model (e.g. core UNIX draft POSIX).
- Current text in acls-04 does that
  - Needs discussion and agreement.

Consensus Items to Resolve
#114 – Clarify Document Goals

- Desirable goal is to provide interoperability across the board.
  - Existing specifications made that difficult because of excessive semantic laxity.
  - Difficult/impossible to get agreement on proper approach post facto.
- We needs to get interoperability for important sub-cases:
  - Core UNIX ACLS & draft POSX ACLs
    - Will need to live with a small set of behavioral variants.
  - Other extensions will get a big improvement, but true interoperability will take more time.

Consensus Items to Resolve
#117 – Better support for Draft POSIX ACLs

- Previous specs had tried to accommodate servers using this model, but ...
  - Did so incompletely for severs.
  - Did nothing for better client support
- Now treated as a set of **OPTIONAL** extensions to the core UNIX ACL model:
  - Support for prohibition on partial satisfaction on ALLOW ACEs
  - Support for default ACL model.
- These required small protocol extensions (new flag bits)