Network File System Version 4
Internet-Draft
Updates: 7530 (if approved)
Intended status: Standards Track
Expires: August 5, 2019

C. Lever, Ed.
Oracle
D. Noveck
NetApp

February 1, 2019

NFS version 4.0 Trunking Update
draft-ietf-nfsv4-mv0-trunking-update-04

Abstract

The file system location-related attribute in NFS version 4.0, fs_locations, informs clients about alternate locations of file systems. An NFS version 4.0 client can use this information to handle migration and replication of server filesystems. This document describes how an NFS version 4.0 client can additionally use this information to discover an NFS version 4.0 server's trunking capabilities. This document updates RFC 7530.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 5, 2019.

Copyright Notice

Table of Contents

1. Introduction

The NFS version 4.0 specification [RFC7530] defines a migration feature that enables the transfer of a file system from one server to another without disruption of client activity. There were a number of issues with the original definition of this feature, now resolved with the publication of [RFC7931].

After a migration event, a client must determine whether state recovery is necessary. To do this, it needs to determine whether the source and destination server addresses represent the same server instance, if the client has already established a lease on the destination server for other file systems, and if the destination server instance has lock state for the migrated file system.

As part of addressing this need, [RFC7931] introduces trunking into NFS version 4.0 along with a trunking detection mechanism. A trunking detection mechanism enables a client to determine whether two distinct network addresses are connected to the same NFS version 4.0 server instance. Without this knowledge, a client unaware of a trunking relationship between paths it is using simultaneously is likely to become confused in ways described in [RFC7530].

NFSv4.1 was defined with an integral means of trunking detection, described in [RFC5661]. NFSv4.0 initially did not have one, with it being added by [RFC7931]. Nevertheless, the use of the concept of server-trunkability is the same in both protocol versions.

File system migration, replication, and referrals are distinct protocol features. However, it is not appropriate to treat each of these features in isolation. For example, client migration recovery processing needs to deal with the possibility of multiple server addresses in a returned fs_locations attribute. In addition, the contents of the fs_locations attribute, which provides both trunking-related and replication information, may change over repeated retrievals, requiring an integrated description of how clients are to deal with such changes. The issues discussed in the current document relate to the interpretation of the fs_locations attribute and to the proper client and server handling of changes in fs_locations attribute values.

Therefore the goals of the current document are:

- To provide NFS version 4.0 with a means of finding addresses trunkable with a given address; i.e., trunking discovery, compatible with the means of trunking detection introduced by [RFC7931]. For an explanation of trunking detection and discovery, see Section 3.

- To describe how NFS version 4.0 clients are to handle the presence of multiple network addresses associated to the same server, when recovering from a replication and migration event.

- To describe how NFS version 4.0 clients are to handle changes in the contents of returned fs_locations attributes, including those

that indicate changes in the responding NFS version 4.0 server's trunking configuration.

The current document pursues these goals by presenting a set of updates to [RFC7530] as summarized in Sections 5 and 6 below.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

Most of the terms related to handling the fs_locations attribute are appropriately defined in Section 5.1 below. However, there are a few terms used outside that context that are explained in this section.

Trunking refers to a situation in which a client uses multiple network addresses communicate with the same server. Trunking was first introduced to NFSv4.0 by [RFC7931]. Regarding network addresses and the handling of trunking we use the following terminology:

- Each NFSv4 server is assumed to have a set of IP addresses to which NFSv4 requests may be sent by clients. These are referred to as the server's network addresses. Access to a specific server network address might involve the use of multiple network ports, since the ports to be used for particular types of connections might be required to be different.

- Clients may establish connections to NFSv4 servers via one of several connection types, supporting the NFSv4 protocol layered on top of an RPC stream transport, as described in [RFC5531], or on top of RPC-over-RDMA, as described in [RFC8166]. The combination of a server network address and a particular connection type is referred to as a "server endpoint".

- Each network address, when combined with a pathname providing the location of a file system root directory relative to the associated server root file handle, defines a file system network access path.

- Two network addresses connected to the same server are said to be server-trunkable. Unlike subsequent NFSv4 minor versions, NFSv4.0

recognizes only a single type of trunking relationship between addresses.

Particularly important is the distinction between trunking detection and trunking discovery. The definitions we present are applicable to all minor versions of NFSv4, but we put particular emphasis on how these terms apply to NFS version 4.0.

- Trunking detection refers to ways of confirming that two unique network addresses are associated with the same NFSv4 server instance. The means available to make this determination depends on the protocol version and, in some cases, on the client implementation.

  In the case of NFS version 4.0, the means to be used are described in [RFC7931] and require use of the Uniform Client String approach to be effective. This is in contrast to later minor versions for which the means of trunking detection are described by [RFC5661].

- Trunking discovery is a process by which an NFSv4 client, accessing one server network address, can obtain other addresses that might be associated with the same server instance. Typically a client builds on a trunking detection facility by providing one or more methods by which candidate addresses are made available to the client, who then uses trunking detection to appropriately filter them.

  Trunking discovery is not discussed in [RFC7530] and no description of it is provided in [RFC7931].

Discussion of the term "replica" is complicated for a number of reasons. Even though the term is used in explaining the issues in [RFC7530] that need to be addressed in the current document, a full explanation of this term requires explanation of related terms connected to the fs_locations attribute, which is provided in Section 5.1 of the current document.

The term is also used in previous documents about NFSv4.0 (i.e., [RFC7530] and [RFC7931]) with a meaning different from that in the current document. In these documents each replica is identified by a single network access path. However, in the current document a set of network access paths which have server-trunkable network addresses and the same root-relative file system pathname are considered to be a single replica with multiple network access paths. Although [RFC7931] enables an NFSv4.0 client to determine whether two network addresses were server-trunkable, it never described these as connected to a single replica, leaving in effect the approach established in [RFC7530].

4. Document Organization

The sections of the current document are divided into four types based on how they relate to the eventual updating of the NFS verion 4.0 specification. Once this update is published, NFS version 4.0 will be specified by multiple documents that need to be read together, until such time as a consolidated replacement specification is produced.

- The base specification [RFC7530].
- The migration-related update [RFC7931].
- This document [RFC-TBD].

The section types are as follows. See Appendix A for a classification of each section of the current document.

- An explanatory section does not contain any material that is meant to update the specification of NFS version 4.0. Such sections may contain explanation about why and how changes are to be done, but do not include any text that is to update [RFC7530] or appear in an eventual consolidated document.
- A replacement section contains text that is to replace and thus supersede text within [RFC7530] and then appear in an eventual consolidated document. The titles of replacement sections indicate what section of [RFC7530] is to be replaced.
- An additional section contains text which, although not replacing anything in [RFC7530], will be part of the specification of NFS version 4.0 and will be expected to be part of an eventual consolidated document. The titles of additional sections provide an indication of where in an updated [RFC7530], the new section would appear.
- An editing section contains some text that replaces text within [RFC7530], although the entire section will not consist of such text and will include other text as well. Such sections make relatively minor adjustments in the existing NFS version 4.0 specification which are expected to be reflected in an eventual consolidated document. Generally such replacement text appears as a quotation, possibly taking the form of an indented set of paragraphs.

5. Changes Within Section 8 of [RFC7530]

Most of the updates to [RFC7530] to provide support for trunking using the fs_locations attribute apply to Section 8 of that document, entitled "Multi-Server Namespace".

- Section 5.1 replaces Section 8.1 of [RFC7530], entitled "Location Attributes". This section has been reorganized and extended to explicitly allow the use of fs_locations to provide trunking-related information that appropriately interacts with the migration, replication and referral features of fs_locations. Terminology used to describe the interactions is added.

- Section 5.2 updates Section 8.4 of [RFC7530], entitled "Uses of Location Information". This section comprises the bulk of the updates. Each paragraph of Section 8.4 and its sub-sections has been reviewed to clarify the provision of trunking-related information using the fs_locations attribute.

  * Section 5.2.1 replaces the introductory material within Section 8.4 of [RFC7530]; i.e., the material within Section 8.4 exclusive of subsections.

  * Section 5.2.2 is to be added as a new sub-section of Section 8.4 before the updated Section 8.4.1 of [RFC7530]. In a consolidated document, it would appear as Section 8.4.1.

  * Section 5.2.3 is to be added as a new sub-section of Section 8.4 before the updated Section 8.4.1 of [RFC7530]. In a consolidated document, it would appear as Section 8.4.2.

  * Section 5.2.4 replaces Section 8.4.1 of [RFC7530], entitled "File System Replication". In a consolidated document, it would appear as Section 8.4.3.

  * Section 5.2.5 replaces Section 8.4.2 of [RFC7530], entitled "File System Migration". In a consolidated document, it would appear as Section 8.4.4.

  * Section 5.2.6 is to be added as a new sub-section of Section 8.4 before Section 8.4.3 of [RFC7530]. In a consolidated document, it would appear as Section 8.4.5, while the existing Section 8.3 would appear as Section 8.4.6.

- Section 5.3 replaces Section 8.5 of [RFC7530], entitled "Location Entries and Server Identity". The last paragraph of the existing section has been removed.

5.1. Updated Section 8.1 of [RFC7530], entitled "Location Attributes"

The fs_locations attribute allows specification of file system locations where the data corresponding to a given file system may be accessed. This attribute represents such file system instances as a server address target (as either a DNS host name representing one or more network addresses, or a single literal network address) together with the path of that file system within the associated single-server namespace. Individual fs_locations entries can express trunkable addresses, locations of file system replicas on other servers, migration targets, or pure referrals.

We introduce the following terminology:

- Trunking is a situation in which multiple network addresses are connected to the same NFS server. Network addresses connected to the same NFS server instance are said to be server-trunkable.

- Trunking detection refers to ways of confirming that two distinct network addresses are connected to the same NFSv4 server instance.

- Trunking discovery is a process by which a client using one network address can obtain other candidate addresses that are server-trunkable with it.

Regarding terminology relating to GETATTR attributes used in trunking discovery and other multi-server namespace features:

- Location attributes include only the fs_locations GETATTR attribute.

- Location entries (fs_location4, defined in [RFC7530] Section 2.2.6) are the individual file system locations in the fs_locations attribute (defined in [RFC7530] Section 2.2.7). A file system location entry designates a set of network addresses to which clients may establish connections. The entry may designate multiple such addresses because the server host name may map to multiple network addresses, and because multiple connection types may be used to communicate with each specified network address. Such addresses provide multiple ways of connecting to a single server.

  Clients use the NFSv4.0 trunking detection mechanism [RFC7931] to confirm that such addresses are connected to the same server. The client can ignore non-confirmed trunking relationships and treat the corresponding addresses as connected to different servers.

- File system location elements are derived from file system location entries. If a file system location entry specifies a network address, there is only a single corresponding location element. When a file system location entry contains a host name, the client resolves the hostname, producing one file system location element for each of the resulting network addresses. Issues regarding the trustworthiness of hostname resolutions are further discussed in Section 7 of the current document.

- All file system location elements consist of a file system location address, which is the network address of an interface to a server, and an fs_name, which is the location of the file system within the server's pseudo-fs.

- If the server has no pseudo-fs and only a single exported file system at the root filehandle, the fs_name may be empty.

5.2. Updates to Section 8.4 of [RFC7530], entitled "Uses of Location Information"

The subsections below provide replacement sections for existing sections within Section 8.4 of [RFC7530] or new sub-sections to be added to that section.

5.2.1. Updated Introduction to Section 8.4 of [RFC7530], entitled "Uses of Location Information"

Together with the possibility of absent file systems, the file system location-bearing attribute fs_locations provides a number of important facilities that enable reliable, manageable, and scalable data access.

When a file system is present on the queried server, this attribute can provide a set of locations that clients may use to access the file system. In the event that server failure, communications problems, or other difficulties make continued access to the file system impossible or otherwise impractical, the returned information provides alternate locations that enable continued access to the file system. Provision of such alternative file system locations is referred to as "replication".

When alternative file system locations are provided, they may represent distinct physical copies of the same file system data or separate NFS server instances that provide access to the same physical file system. Another possible use of the provision of multiple file system location entries is trunking, wherein the file system location entries do not in fact represent different servers but rather are distinct network paths to the same server.

A client may use file system location elements simultaneously to provide higher-performance access to the target file system. This can be done using trunking, although the use of multiple replicas simultaneously is possible. To enable simultaneous access, the client utilizes trunking detection and/or discovery, further described in Section 5.2.2 of the current document, to determine a set of network paths that are server-trunkable with the one currently being used to access the file system. Once this determination is made, requests may be routed across multiple paths, using the existing state management mechanism.

Multiple replicas may also be used simultaneously, typically used when accessing read-only datasets. In this case, each replica requires its own state management. The client performs multiple file opens to read the same file content from multiple replicas.

When a file system is present and subsequently becomes absent, clients can be given the opportunity to have continued access to their data at an alternative file system location. Transfer of the file system contents to the new file system location is referred to as "migration". The client's responsibilities in dealing with this transition depend on the specific nature of the new access path as well as how and whether data was in fact migrated. See Sections 5.2.5 and 5.2.6 of the current document for details.

The fs_locations attribute can designate one or more remote file system locations in place of an absent file system. This is known as a "referral". A particularly important case is that of a "pure referral", in which the absent file system has never been present on the NFS server. Such a referral is a means by which a file system located on one server can redirect clients to file systems located on other servers, thus enabling the creation of a multi-server namespace.

Because client support for the fs_locations attribute is OPTIONAL, a server may (but is not required to) take action to hide migration and referral events from such clients by acting as a proxy, for example.

5.2.2. New Sub-section of Section 8.4 of [RFC7530], to be entitled "Trunking Discovery and Detection"

Trunking is a situation in which multiple distinct network addresses are associated with the same NFS server instance. As a matter of convenience, we say that two network addresses connected to the same NFS server instance are server-trunkable. Section 5.4 of [RFC7931] explains why NFSv4 clients need to be aware of NFS server identity to manage lease and lock state effectively when multiple connections to the same server exist.

Trunking detection refers to a way for an NFSv4 client to confirm that two independently acquired network addresses are connected to the same NFSv4 server. Section 5.8 of [RFC7931] describes an OPTIONAL means by which it can be determined if two network addresses correspond to the same NFSv4.0 server instance. Without trunking detection, an NFSv4.0 client has no other way to confirm that two network addresses are server-trunkable.

In the particular context of NFS version 4.0, trunking detection requires that the client support the Uniform Client ID String approach (UCS), described in Section 5.6 of [RFC7931]. Any NFSv4.0 client that supports migration or trunking detection needs to present a Uniform Client ID String to all NFSv4.0 servers. If it does not do so, it will be unable to perform trunking detection.

Trunking discovery is the process by which an NFSv4 client using a host name or one of an NFSv4 server's network addresses can obtain other candidate network addresses that are trunkable with it; i.e., a set of addresses that might be connected to the same NFSv4 server instance. An NFSv4.0 client can discover server-trunkable network addresses in a number of ways:

- An NFS server's host name is provided either at mount time or in a returned file system location entry. A DNS query of this host name can return more than one network address. The returned network addresses are candidates for trunking.

- Location entries returned in an fs_locations attribute can specify network addresses. These network addresses are candidates for trunking.

When there is a means of trunking detection available, an NFSv4.0 client can confirm that a set of network addresses correspond to the same NFSv4.0 server instance and thus any of them can be used to access that server.

5.2.3. New Sub-section of Section 8.4 of [RFC7530], to be entitled "Location Attributes and Connection Type Selection"

NFS version 4.0 may be implemented using a number of different types of connections:

Stream connections may be used to provide RPC service, as described in [RFC5531].

RDMA-capable connections may be used to provide RPC service, as described in [RFC8166].

Because of the need to support multiple connections, clients face the issue of determining the proper connection type to use when establishing a connection to a server network address. The fs_locations attribute provides no information to support connection type selection. As a result, clients supporting multiple connection types need to attempt to establish a connection on various connection types allowing it to determine, via a trial-and-error approach, which connection types are supported.

If a client strongly prefers one connection type, it can perform these attempts serially in order of declining preference. Once there is a successful attempt, the established connection can be used. Note that with this approach, network partitions can result in a sequence of long waits for a successful connection.

To avoid waiting when there is at least one viable network path available, simultaneous attempts to establish multiple connection types are possible. Once a viable connection is established, the client discards less-preferred connections.

5.2.4. Updated Section 8.4.1 of [RFC7530], entitled "File System Replication and Trunking"

On first access to a file system, the client should obtain the value of the set of alternative file system locations by interrogating the fs_locations attribute. Trunking discovery and/or detection can then be applied to the file system location entries to separate the candidate server-trunkable addresses from the replica addresses that provide alternative locations of the file system. Server-trunkable addresses may be used simultaneously to provide higher performance through the exploitation of multiple paths between client and target file system.

In the event that server failures, communications problems, or other difficulties make continued access to the current file system impossible or otherwise impractical, the client can use the alternative file system locations as a way to maintain continued access to the file system. See Section 5.2.6 of the current document for more detail.

5.2.5. Updated Section 8.4.2 of [RFC7530], entitled "File System Migration"

When a file system is present and becomes absent, clients can be given the opportunity to have continued access to their data at an alternative file system location specified by the fs_locations attribute. Typically, a client will be accessing the file system in question, get an NFS4ERR_MOVED error, and then use the fs_locations

attribute to determine the new location of the data. See Section 5.2.6 of the current document for more detail.

Such migration can help provide load balancing or general resource reallocation. The protocol does not specify how the file system will be moved between servers. It is anticipated that a number of different server-to-server transfer mechanisms might be used, with the choice left to the server implementer. The NFSv4 protocol specifies the method used to communicate the migration event between client and server.

When the client receives indication of a migration event via an NFS4ERR_MOVED error, data propagation to the destination server must have already occurred. Once the client proceeds to access the alternate file system location, it must see the same data. Where file systems are writable, a change made on the original file system must be visible on all migration targets. Where a file system is not writable but represents a read-only copy (possibly periodically updated) of a writable file system, similar requirements apply to the propagation of updates. Any change visible in the original file system must already be effected on all migration targets, to avoid any possibility that a client, in effecting a transition to the migration target, will see any reversion in file system state.

5.2.6. **New Sub-section of Section 8.4 of [RFC7530], to be entitled** "Interaction of Trunking, Migration, and Replication"

When the set of network addresses on a server change in a way that would affect a file system location attribute, there are several possible outcomes for clients currently accessing that file system. NFS4ERR_MOVED is returned only when the server cannot satisfy a request from the client, whether because the file system has been migrated to a different server, is only accessible at a different trunked address on the same server, or some other reason. In the cases 1 and 2 below, NFS4ERR_MOVED is not returned.

1. When the list of network addresses is a superset of that previously in effect, there is no need for migration or any other sort of client adjustment. Nevertheless, the client is free to use an additional address in the replacement list if that address provides another path to the same server. Or, the client may treat that address as it does a replica, to be used if current server addresses become unavailable.

2. When the list of network addresses is a subset of that previously in effect, immediate action is not needed if an address missing in the replacement list is not currently in use by the client. The client should avoid using that address to access that file

system in the future, whether the address is for a replica or an additional path to the server being used.

3. When an address being removed is one of a number of paths to the current server, the client may continue to use it until NFS4ERR_MOVED is received. This is not considered a migration event unless the last available path to the server has become unusable.

When migration does occur, multiple addresses may be in use on the server previous to migration and multiple addresses may be available for use on the destination server.

With regard to the server in use, it may be that return of NFS4ERR_MOVED indicates that a particular network address is no longer to be used, without implying that migration of the file system to a different server is needed. Clients should not conclude that migration has occurred until confirming that all network addresses known to be associated with that server are not usable.

It should be noted that the need to defer this determination is not absolute. If a client is not aware of all network addresses for any reason, it may conclude that migration has occurred when it has not and treat a switch to a different server address as if it were a migration event. This is harmless since the use of the same server via a new address will appear as a successful instance of Transparent State Migration.

Although significant harm cannot arise from this misapprehension, it can give rise to disconcerting situations. For example, if a lock has been revoked during the address shift, it will appear to the client as if the lock has been lost during migration. When such a lock is lost, it is the responsibility of the destination server to provide for its recovery via the use of an fs-specific grace period.

With regard to the destination server, it is desirable for the client to be aware of all valid network addresses that can be used to access the destination server. However, there is no need for this to be done immediately. Implementations can process the additional file system location elements in parallel with normal use of the first valid file system location entry found to access the destination.

Because a file system location attribute may include entries relating to the current server, the migration destination, and possible replicas to use, scanning for available network addresses that might be trunkable with addresses the client has already seen could potentially be a long process. To keep this process as short as possible, servers that provide information about trunkable network

paths are REQUIRED to place file system location entries that represent addresses usable with the current server or a migration target before those associated with replicas.

This ordering allows a client to cease scanning for trunkable file system location entries once it encounters a file system location element whose fs_name differs from the current fs_name, or whose address is not server-trunkable with the one it is currently using. Although the possibility exists that a client might prematurely cease scanning for trunkable addresses when receiving a location attribute from an older server that does not follow the ordering constraint above, the harm is expected to be limited since such servers would not be expected to present information about trunkable server access paths.

5.3. Updated Section 8.5 of [RFC7530], entitled "Location Entries and Server Identity Update"

As mentioned above, a single file system location entry may have a server address target in the form of a DNS host name that resolves to multiple network addresses; it is also possible for multiple file system location entries to have their own server address targets that reference the same server.

When server-trunkable addresses for a server exist, the client may assume that for each file system in the namespace of a given server network address, there exist file systems at corresponding namespace locations for each of the other server-trunkable network addresses. It may do this even in the absence of explicit listing in fs_locations. Such corresponding file system locations can be used as alternative locations, just as those explicitly specified via the fs_locations attribute.

If a single file system location entry designates multiple server IP addresses, the client should choose a single one to use. When two server addresses are designated by a single file system location entry and they correspond to different servers, this normally indicates some sort of misconfiguration. The client should avoid using such file system location entries when alternatives are available. When they are not, the client should pick one of the IP addresses and use it, without using others that are not directed to the same server.

6. Updates to [RFC7530] Outside Section Eight

Since the existing description of NFS4ERR_MOVED in Section 13.1.2.4 of [RFC7530] does not take proper account of trunking, it needs to be

modified by replacing the first two sentences of the description with the following material:

> The file system that contains the current filehandle object cannot be accessed using the current network address. It may be accessible using other network addresses connected to the same server, it may have been relocated to another server, or it may never have been present.

7. Security Considerations

The Security Considerations section of [RFC7530] needs the additions below to properly address some aspects of trunking discovery, referral, migration, and replication.

> The possibility that requests to determine the set of network addresses corresponding to a given server might be interfered with or have their responses corrupted needs to be taken into account.

- When DNS is used to convert NFS server host names to network addresses and DNSSEC [RFC4033] is not available, the validity of the network addresses returned cannot be relied upon. However, when the client uses RPCSEC_GSS [RFC7861] to access NFS servers, it is possible for mutual authentication to detect invalid server addresses. Other forms of transport layer security (e.g., [RFC8446]) can also offer strong authentication of NFS servers.

- Fetching file system location information SHOULD be performed using RPCSEC_GSS with integrity protection, as previously explained in the Security Considerations section of [RFC7530]. Making a request of this sort without using strong integrity protection permits corruption during transit of returned file system location information. The client implementer needs to recognize that using such information to access an NFS server without use of RPCSEC_GSS (e.g., by using AUTH_SYS as defined in [RFC5531]) can result in the client interacting with an unverified network address that is posing as an NFSv4 server.

- Despite the fact that it is a REQUIREMENT of [RFC7530] that "implementations" provide "support" for the use of RPCSEC_GSS, it cannot be assumed that use of RPCSEC_GSS is always possible between any particular client-server pair.

- Returning only network addresses to a client that has no trusted DNS resolution service can hamper its ability to use RPCSEC_GSS.

Therefore an NFSv4 server SHOULD present file system location entries that correspond to file systems on other servers using only host names. This enables the client to interrogate the fs_locations on the destination server to obtain trunking information (as well as replica information) using RPCSEC_GSS with integrity, validating the host name provided while assuring that the response has not been corrupted.

When RPCSEC_GSS is not available on an NFS server, returned file system location information is subject to corruption during transit and cannot be relied upon. In the case of a client being directed to another server after NFS4ERR_MOVED, this could vitiate the authentication provided by the use of RPCSEC_GSS on the destination. Even when RPCSEC_GSS authentication is available on the destination, this server might validly represent itself as the server to which the client was erroneously directed. Without a way to decide wether the server is a valid one, the client can only determine, using RPCSEC_GSS, that the server corresponds to the host name provided, with no basis for trusting that server. The client should not use such unverified file system location entries as a basis for migration, even though RPCSEC_GSS might be available on the destination server.

When a file system location attribute is fetched upon connecting with an NFSv4 server, it SHOULD, as stated above, be done using RPCSEC_GSS with integrity protection.

When file system location information cannot be protected in transit, the client can subject it to additional filtering to prevent the client from being inappropriately directed. For example, if a range of network addresses can be determined that assure that the servers and clients using AUTH_SYS are subject to appropriate constraints (such as physical network isolation and the use of administrative controls within the operating systems), then network adresses in this range can be used, with others discarded or restricted in their use of AUTH_SYS.

When neither integrity protection nor filtering is possible, it is best for the client to ignore trunking and replica information or simply not fetch the file system location information for these purposes.

To summarize considerations regarding the use of RPCSEC_GSS in fetching file system location information, consider the following recommendations for requests to interrogate location information, with interrogation approaches on the referring and destination servers arrived at separately:

- The use of RPCSEC_GSS with integrity protection is RECOMMENDED in all cases, since the absence of integrity protection exposes the client to the possibility of the results being modified in transit.

- The use of requests issued without RPCSEC_GSS (e.g., using AUTH_SYS), while undesirable, might be unavoidable in some cases. Where the use of returned file system location information cannot be avoided, it should be subject to filtering to eliminate untrusted network addresses. The specifics will vary depending on the degree of network isolation and whether the request is to the referring or destination servers.

Privacy considerations relating to uniform client strings (UCS) vs. non-uniform client strings (non-UCS), discussed in Section 5.6 of [RFC7931], are also applicable to their usage for trunking detection in NFS version 4.0.

8. IANA Considerations

This document does not require actions by IANA.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 5531, DOI 10.17487/RFC5531, May 2009, <https://www.rfc-editor.org/info/rfc5531>.

[RFC7530] Haynes, T., Ed. and D. Noveck, Ed., "Network File System (NFS) Version 4 Protocol", RFC 7530, DOI 10.17487/RFC7530, March 2015, <https://www.rfc-editor.org/info/rfc7530>.

[RFC7931] Noveck, D., Ed., Shivam, P., Lever, C., and B. Baker, "NFSv4.0 Migration: Specification Update", RFC 7931, DOI 10.17487/RFC7931, July 2016, <https://www.rfc-editor.org/info/rfc7931>.

[RFC8166] Lever, C., Ed., Simpson, W., and T. Talpey, "Remote Direct Memory Access Transport for Remote Procedure Call Version 1", RFC 8166, DOI 10.17487/RFC8166, June 2017, <https://www.rfc-editor.org/info/rfc8166>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC5661] Shepler, S., Ed., Eisler, M., Ed., and D. Noveck, Ed., "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 5661, DOI 10.17487/RFC5661, January 2010, <https://www.rfc-editor.org/info/rfc5661>.

[RFC7861] Adamson, A. and N. Williams, "Remote Procedure Call (RPC) Security Version 3", RFC 7861, DOI 10.17487/RFC7861, November 2016, <https://www.rfc-editor.org/info/rfc7861>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Appendix A. Section Classification

All sections of the current document are considered explanatory with the following exceptions.

- Sections 5.1 and 5.2.1 are replacement sections.
- Section 5.2.2 is an additional section.
- Sections 5.2.4 and 5.2.5 are replacement sections.
- Section 5.2.6 is an additional section.
- Section 5.3 is a replacement section.
- Section 6 is an editing section.
- Section 7 is an additional section.

Acknowledgments

The authors wish to thank Andy Adamson, who wrote the original version of this document. All the innovation in this document is the result of Andy's work, while mistakes are best ascribed to the current authors.

The editor wishes to thank Greg Marsden for his support of this work, and Robert Thurlow for his review and suggestions.

Special thanks go to Transport Area Director Spencer Dawkins, NFSV4 Working Group Chairs Spencer Shepler and Brian Pawlowski, and NFSV4 Working Group Secretary Thomas Haynes for their ongoing support. We are also grateful for the thorough review of this document by Benjamin Kaduk and Ben Campbell.

Authors' Addresses

Charles Lever (editor)
Oracle Corporation
United States of America

Email: chuck.lever@oracle.com

David Noveck
NetApp
United States of America

Email: davenoveck@gmail.com