Multiple NFSv4 Domain Namespace Deployment Guidelines
draft-ietf-nfsv4-multi-domain-fs-reqs-06

Abstract

   This document discusses issues relevant to the deployment of the
   NFSv4 protocols in situations allowing for the construction of an
   NFSv4 file namespace supporting the use of multiple NFSv4 domains and
   utilizing multi-domain capable file systems. Also described are
   constraints on name resolution and security services appropriate to
   the administration of such a system. Such a namespace is a suitable
   way to enable a Federated File System supporting the use of multiple
   NFSv4 domains.

Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC2119].

Status of This Memo

Table of Contents

1. Introduction

    An NFSv4 domain is defined as a set of users and groups named by a
    particular domain using the NFSv4 name@domain syntax. This includes
    NFSv4.0 [RFC7530], NFSv4.1 [RFC5661], and minor versions yet to be
    published. Often, a computer which acts as an NFSv4 client and

always acts on behalf of users belonging to a particular NFSv4 domain
is thought of a part of that NFSv4 domain. Similarly, a computer
acting as an NFSv4 server that is only aware of users within a
particular NFSv4 domain may be thought of as part of that NFSv4
domain.

In this document, the term "multi-domain" always refers to multiple
NFSv4 domains.

The Federated File System (FedFS) [RFC5716] describes the
requirements and administrative tools to construct a uniform NFSv4
file server based namespace that is capable of spanning a whole
enterprise and that is easy to manage.

The FedFS is the standardized method of constructing and
administrating an enterprise-wide NFSv4 filesystem, and so is
referenced in this document. The issues with multi-domain
deployments described in this document apply to all multi-domain
deployments, whether they are run as a FedFS or not.

Stand-alone NFSv4 domain deployments can be run in many ways. While
a FedFS can be run within all stand-alone NFSv4 domain configurations
some of these configurations (Section 4) are not compatible with
joining a multi-domain FedFS namespace.

Multi-domain deployments require support for global identities in
name services and security services, and file systems capable of the
on-disk representation of identities belonging to multiple NFSv4
domains. Typically, stand-alone NFSv4 domain deployments only
provide support for identities belonging to a single NFSv4 domain.

This document describes administration-related constraints applying
to the deployment of the NFSv4 protocols in environments supporting
the construction of an NFSv4 file system namespace supporting the use
of multiple NFSv4 domains and utilizing multi-domain capable file
systems. Also described are constraints regarding the name
resolution and security services appropriate to such a deployment.
Such a namespace is a suitable way to enable a Federated File System
supporting the use of multiple NFSv4 domains.

2. Terminology

   Name Service: Facilities that provides the mapping between {NFSv4
   domain, group or user name} and the appropriate local
   representation of identity. Also includes facilities providing
   mapping between a security principal and local representation of
   identity. Can be applied to global identities or principals from

within local and remote domains. Often provided by a Directory
Service such as LDAP.

Name Service Switch (nsswitch): a facility in provides a variety
of sources for common configuration databases and name resolution
mechanisms.

Domain: This term is used in multiple contexts where it has
different meanings. Definitions of "nfsv4 domain" and "multi-
domain" have already appeared above in Section 1. Below we
provide other specific definitions used this document.

> DNS domain: a set of computers, services, or any internet
> resource identified by an DNS domain name [RFC1034].
>
> Security realm or domain: a set of configured security
> providers, users, groups, security roles, and security policies
> running a single security protocol and administered by a single
> entity, for example a Kerberos realm.
>
> FedFS domain: A file namespace that can cross multiple shares
> on multiple file servers using file-access protocols such as
> NFSv4. A FedFS domain is typically a single administrative
> entity, and has a name that is similar to a DNS domain name.
> Also known as a Federation.
>
> Administrative domain: a set of users, groups, computers, and
> services administered by a single entity. Can include multiple
> DNS domains, NFSv4 domains, security domains, and FedFS
> domains.

Local representation of identity: A representation of a user or a
group of users capable of being stored persistently within a file
system. Typically such representations are identical to the form
in which users and groups are represented within internal server
API's. Examples are numeric id's such as a uidNumber (UID),
gidNumber (GID) [RFC2307], or a Windows Security Identifier (SID)
[CIFS]. In some case the identifier space for user and groups
overlap, requiring anyone using such an id to know a priori
whether the identifier is for a user or a group.

Global identity: An on-the-wire globally unique form of identity
that can be mapped to a local representation. For example, the
NFSv4 name@domain or the Kerberos principal@REALM.

Multi-domain capable filesystem: A local filesystem that uses a
local ID form that can represent NFSv4 identities from multiple
domains.

Principal: an RPCSEC_GSS [RFC2203] authentication identity.
Usually, but not always, a user; rarely, if ever, a group;
sometimes a host or server.

Authorization Context: A collection of information about a
principal such as username, userID, group membership, etcetera
used in authorization decisions.

Stringified UID or GID: NFSv4 owner and group strings that consist
of decimal numeric values with no leading zeros, and which do not
contain an '@' sign. See Section 5.9 "Interpreting owner and
owner_group" [RFC5661].

3. Identity Mapping

3.1. NFSv4 Server Identity Mapping

NFSv4 servers deal with two kinds of identities: authentication
identities (referred to here as "principals") and authorization
identities ("users" and "groups" of users). NFSv4 supports multiple
authentication methods, each authenticating an "initiator principal"
(typically representing a user) to an "acceptor principal" (always
corresponding to the NFSv4 server). NFSv4 does not prescribe how to
represent authorization identities on file systems. All file access
decisions constitute "authorization" and are made by NFSv4 servers
using authorization context information and file metadata related to
authorization, such as a file's access control list (ACL).

NFSv4 servers therefore must perform two kinds of mappings:

1. Auth-to-authz: A mapping between the authentication identity and
    the authorization context information.

2. Wire-to-disk: A mapping between the on-the-wire authorization
    identity representation and the on-disk authorization identity
    representation.

A Name Service such as LDAP often provides these mappings.

Many aspects of these mappings are entirely implementation specific,
but some require multi-domain capable name resolution and security
services in order to interoperate in a multi-domain environment.

NFSv4 servers use these mappings for:

1. File access: Both the auth-to-authz and the wire-to-disk mappings
    may be required for file access decisions.

2. Meta-data setting and listing: The auth-to-authz mapping is
    usually required to service file metadata setting or listing
    requests such as ACL or unix permission setting or listing. This
    mapping is needed because NFSv4 messages use identity
    representations of the form name@domain which normally differs
    from the server's local representation of identity.

3.2. NFSv4 Client Identity Mapping

   A client setting the owner or group attribute will often need access
   to identity mapping services. This is because API's within the
   client will specify the identity in a local form (e.g UNIX using a
   uid/gid) so that when stringified id's cannot be used, the id must be
   converted to a global form.

   A client obtaining values for the owner or group attributes will
   similarly need access to identity mapping services. This is because
   the client API will need these attributes in a local form, as above.
   As a result name services need to be available to convert the global
   identity to a local form.

   Note that each of these situations arises because client-side API's
   require a particular local identity representation. The need for
   mapping services would not arise if the clients could use the global
   representation of identity directly.

4. Stand-alone NFSv4 Domain Deployment Examples

   In order to service as many environments as possible, the NFSv4
   protocol is designed to allow administrators freedom to configure
   their NFSv4 domains as they please.

   Stand-alone NFSv4 domains can be run in many ways. Here we list some
   stand-alone NFSv4 domain deployment examples focusing on the NFSv4
   server's use of name service mappings (Section 3.1) and security
   services deployment to demonstrate the need for some multiple NFSv4
   domain constraints to the NFSv4 protocol, name service configuration,
   and security service choices.

   Because all on-disk identities participating in a stand-alone NFSv4
   domain belong to the same NFSv4 domain, stand-alone NFSv4 domain
   deployments have no requirement for exporting multi-domain capable
   file systems.

   These examples are for a NFSv4 server exporting a POSIX UID/GID based
   file system, a typical deployment. These examples are listed in the
   order of increasing NFSv4 administrative complexity.

4.1. AUTH_SYS with Stringified UID/GID

   This example is the closest NFSv4 gets to being run as NFSv3.

   File access: The AUTH_SYS RPC credential provides a UID as the
   authentication identity, and a list of GIDs as authorization context
   information. File access decisions require no name service
   interaction as the on-the-wire and on-disk representation are the
   same and the auth-to-authz UID and GID authorization context
   information is provided in the RPC credential.

   Meta-data setting and listing: When the NFSv4 clients and servers
   implement a stringified UID/GID scheme, where a stringified UID or
   GID is used for the NFSv4 name@domain on-the-wire identity, then a
   name service is not required for file metadata listing as the UID or
   GID can be constructed from the stringified form on the fly by the
   server.

4.2. AUTH_SYS with name@domain

   Another possibility is to express identity using the form
   'name@domain', rather than using a stringified UID/GID scheme for
   file metadata setting and listing.

   File access: This is the same as in Section 4.1.

   Meta-data setting and listing: The NFSv4 server will need to use a
   name service for the wire-to-disk mappings to map between the on-the-
   wire name@domain syntax and the on-disk UID/GID representation.
   Often, the NFSv4 server will use the nsswitch interface for these
   mappings. A typical use of the nsswitch name service interface uses
   no domain component, just the uid attribute [RFC2307] (or login name)
   as the name component. This is no issue in a stand-alone NFSv4
   domain deployment as the NFSv4 domain is known to the NFSv4 server
   and can combined with the login name to form the name@domain syntax
   after the return of the name service call.

4.3. RPCSEC_GSS with name@domain

   RPCSEC_GSS uses GSS-API [RFC2743] security mechanisms to securely
   authenticate users to servers. The most common mechanism is Kerberos
   [RFC4121].

   This final example adds the use of RPCSEC_GSS with the Kerberos 5 GSS
   security mechanism.

   File Access: The forms of GSS principal names are mechanism-specific.
   For Kerberos these are of the form principal@REALM. Sometimes

authorization context information is delivered with authentication,
but this cannot be counted on. Authorization context information not
delivered with authentication has timely update considerations (i.e.,
generally it's not possible to get a timely update). File access
decisions therefore require a wire-to-disk mapping of the GSS
principal to a UID, and an auth-to-authz mapping to obtain the list
of GIDs as the authorization context.

Implementations must never blindly drop a Kerberos REALM name from a
Kerberos principal name to obtain a POSIX username, but they may be
configured to do so for specific REALMs.

Meta-data setting and listing: This is the same as in Section 4.2.

5. Multi-domain Constraints to the NFSv4 Protocol

Joining NFSv4 domains under a single file namespace imposes slightly
on the NFSv4 administration freedom. Here we describe the required
constraints.

5.1. Name@domain Constraints

NFSv4 uses a syntax of the form "name@domain" as the on-the-wire
representation of the "who" field of an NFSv4 access control entry
(ACE) for users and groups. This design provides a level of
indirection that allows NFSv4 clients and servers with different
internal representations of authorization identity to interoperate
even when referring to authorization identities from different NFSv4
domains.

Multi-domain capable sites need to meet the following requirements in
order to ensure that NFSv4 clients and servers can map between
name@domain and internal representations reliably. While some of
these constraints are basic assumptions in NFSv4.0 [RFC7530] and
NFSv4.1 [RFC5661], they need to be clearly stated for the multi-
domain case.

o The NFSv4 domain portion of name@domain MUST be unique within the
   multi-domain namespace. See [RFC5661] section 5.9 "Interpreting
   owner and owner_group" for a discussion on NFSv4 domain
   configuration.

o The name portion of name@domain MUST be unique within the
   specified NFSv4 domain.

Due to UID and GID collisions, stringified UID/GIDs MUST NOT be used
in a multi-domain deployment. This means that multi-domain-capable
servers MUST reject requests that use stringified UID/GIDs.

5.1.1. NFSv4 Domain and DNS Services

Here we address the relationship between NFSv4 domain name and DNS
domain name in a multi-domain deployment.

The definition of an NFSv4 domain name needs clarification to work in
a multi-domain file system namespace. Section 5.9 [RFC5661] loosely
defines the NFSv4 domain name as a DNS domain name. This loose
definition for the NFSv4 domain is a good one, as DNS domain names
are globally unique. As noted above in Section 5.1, any choice of
NFSv4 domain name can work within a stand-alone NFSv4 domain
deployment whereas the NFSv4 domain is required to be unique in a
multi-domain deployment.

A typical configuration is that there is a single NFSv4 domain that
is served by a single DNS domain. In this case the NFSv4 domain name
can be the same as the DNS domain name.

An NFSv4 domain can span multiple DNS domains. In this case, one of
the DNS domain names can be chosen as the NFSv4 domain name.

Multiple NFSv4 domains can also share a DNS domain. In this case,
only one of the NFSv4 domains can use the DNS domain name, the other
NFSv4 domains must choose another unique NFSv4 domain name.

5.1.2. NFSv4 Domain and Name Services

As noted above in Section 5.1, each name@domain is unique across the
multi-domain namespace and maps, on each NFSv4 server, to the local
representation of identity used by that server. Typically, this
representation consists of an indication of the particular domain
combined with the uid/gid corresponding to the name component. To
support such an arrangement, each NFSv4 domain needs to have a single
name resolution service capable of converting the names defined
within the domain to the corresponding local representation.

5.2. RPC Security Constraints

As described in [RFC5661] section 2.2.1.1 "RPC Security Flavors":

        NFSv4.1 clients and servers MUST implement RPCSEC_GSS.
        (This requirement to implement is not a requirement
        to use.) Other flavors, such as AUTH_NONE, and AUTH_SYS,
        MAY be implemented as well.

The underlying RPCSEC_GSS security mechanism used in a multi-domain
namespace is REQUIRED to employ a method of cross NFSv4 domain trust
so that a principal from a security service in one NFSv4 domain can

be authenticated in another NFSv4 domain that uses a security service
with the same security mechanism. Kerberos is an example of such a
security services.

The AUTH_NONE security flavor can be useful in a multi-domain
deployment to grant universal access to public data without any
credentials.

The AUTH_SYS security flavor uses a host-based authentication model
where the weakly authenticated host (the NFSv4 client) asserts the
user's authorization identities using small integers, uidNumber, and
gidNumber [RFC2307], as user and group identity representations.
Because this authorization ID representation has no domain component,
AUTH_SYS can only be used in a namespace where all NFSv4 clients and
servers share an [RFC2307] name service. A shared name service is
required because uidNumbers and gidNumbers are passed in the RPC
credential; there is no negotiation of namespace in AUTH_SYS.
Collisions can occur if multiple name services are used, so AUTH_SYS
MUST NOT be used in a multi-domain file system deployment.

While the AUTH_SYS security mechanism can not be used (indeed,
AUTH_SYS is obsolete and of limited use for all of NFS), RPCSEC_GSSv3
[I-D.rpcsec-gssv3] can completely replace all uses of AUTH_SYS in a
multi-domain file system. Like AUTH_SYS, and unlike RPCSEC_GSSv1/2,
RPCSEC_GSSv3 allows the client to assert and contribute knowledge of
the user process' authorization context.

5.2.1. NFSv4 Domain and Security Services

As noted above in Section 5.2, caveat AUTH_NULL, multiple NFSv4
domain security services are RPCSEC_GSS based with the Kerberos 5
security mechanism being the most commonly (and as of this writing,
the only) deployed service.

A single Kerberos 5 security service per NFSv4 domain with the upper
case NFSv4 domain name as the Kerberos 5 REALM name is a common
deployment.

Multiple security services per NFSv4 domain is allowed, and brings
the issue of mapping multiple Kerberos 5 principal@REALMs to the same
local ID. Methods of achieving this are beyond the scope of this
document.

6. Resolving Multi-domain Authorization Information

When an RPCSEC_GSS principal is seeking access to files on an NFSv4
server, after authenticating the principal, the server must obtain in
a secure manner the principal's authorization context information

from an authoritative source such as the name service in the
principal's NFSv4 domain.

In the stand-alone NFSv4 domain case where the principal is seeking
access to files on an NFSv4 server in the principal's home NFSv4
domain, the server administrator has knowledge of the local policies
and methods for obtaining the principal's authorization information
and the mappings to local representation of identity from an
authoritative source. E.g., the administrator can configure secure
access to the local NFSv4 domain name service.

In the multi-domain case where a principal is seeking access to files
on an NFSv4 server not in the principal's home NFSv4 domain, the
NFSv4 server may be required to contact the remote name service in
the principals NFSv4 domain. In this case there is no assumption of:

o Remote name service configuration knowledge.

o The syntax of the remote authorization context information
   presented to the NFSv4 server by the remote name service for
   mapping to a local representation.

There are several methods the NFSv4 server can use to obtain the
NFSv4 domain authoritative authorization information for a remote
principal from an authoritative source. While any detail is beyond
the scope of this document, some general methods are listed here.

1. A mechanism specific GSS-API authorization payload containing
    credential authorization data such as a "privilege attribute
    certificate" (PAC) [PAC] or a "general PAD" (PAD)
    [I-D.sorce-krbwg-general-pac]. This is the preferred method as
    the payload is delivered as part of GSS-API authentication,
    avoids requiring any knowledge of the remote authoritative
    service configuration, and its syntax is well known.

2. When there is a security agreement between the local and remote
    NFSv4 domain name services plus regular update data feeds, the
    NFSv4 server local NFSv4 domain name service can be authoritative
    for principal's in the remote NFSv4 domain. In this case, the
    NFSv4 server makes a query to it's local NFSv4 domain name
    service just as it does when servicing a local domain principal.
    While this requires detailed knowledge of the remote NFSv4
    domains name service for the update data feeds, the authorization
    context information presented to the NFSv4 server is in the same
    form as a query for a local principal.

3. An authenticated direct query from the NFSv4 server to the
    principal's NFSv4 domain authoritative name service. This

        requires the NFSv4 server to have detailed knowledge of the
        remote NFSv4 domain's authoritative name service and detailed
        knowledge of the syntax of the resultant authorization context
        information.

7. Stand-alone Examples and Multiple NFSv4 Domain Namespaces

   Revisiting the stand-alone (Section 4) NFSv4 domain deployment
   examples, we note that due to the use of AUTH_SYS, neither
   Section 4.1 nor Section 4.2 configurations are suitable for multi-
   domain deployments.

   The Section 4.3 configuration example can participate in a multi-
   domain namespace deployment if:

   o The NFSv4 domain name is unique across the namespace.

   o All exported file systems are multi-domain capable.

   o A secure method is used to resolve remote NFSv4 domain principals
      authorization information from an authoritative source.

8. Security Considerations

   This RFC discusses security throughout. All the security
   considerations of the relevant protocols, such as NFSv4.0 [RFC7530],
   NFSv4.1 [RFC5661], RPCSEC_GSS [RFC2203], GSS-API [RFC4121], LDAP
   [RFC4511], and others, apply.

   Authentication and authorization across administrative domains
   presents security considerations, most of which are treated
   elsewhere, but we repeat some of them here:

   o latency in propagation of revocation of authentication credentials

   o latency in propagation of revocation of authorizations

   o latency in propagation of granting of authorizations

   o complications in establishing a foreign domain's users' complete
      authorization context: only parts may be available to servers

   o privacy considerations in a federated environment

   Most of these are security considerations of the mechanisms used to
   authenticate users to servers and servers to users, and of the
   mechanisms used to evaluate a user's authorization context. We don't

treat them fully here, but implementors should study the protocols in
question to get a more complete set of security considerations.

Note that clients/users may also need to evaluate a server's
authorization context when using labeled security [I-D.NFSv4.2]
(e.g., is the server authorized to handle content at a given security
level, for the given compartments). Even when not using labeled
security, since there could be many realms (credential issuer) for a
given server, it's important to verify that the server a client is
talking to has a credential for the name the client has for the
server, and that that credential's issuer (i.e., its realm) is
allowed to issue it. Usually the service principle realm
authorization function is implemented by the security mechanism, but
the implementor should check this.

Implementors may be tempted to assume that realm (or "issuer") and
NFSv4 domain are roughly the same thing, but they are not.
Configuration and/or lookup protocols (such as LDAP) and associated
schemas are generally required in order to evaluate a user
principal's authorization context. In the simplest scheme a server
has access to a database mapping all known principal names to
usernames whose authorization context can be evaluated using
operating system interfaces that deal in usernames rather than
principal names.

9. IANA Considerations

There are no IANA considerations in this document.

10. References

10.1. Normative References

[I-D.NFSv4.2]
           Haynes, T., "NFS Version 4 Minor Version 2", draft-ietf-
           nfsv4-minorversion2-36 (Work In Progress), April 2015.

[I-D.rpcsec-gssv3]
           Adamson, W. and N. Williams, "Remote Procedure Call (RPC)
           Security Version 3", draft-ietf-nfsv4-rpcsec-gssv3-12
           (Work In Progress), July 2015.

[RFC1034] Mockapetris, P., "DOMAIN NAMES - CONCEPTS AND FACILITIES",
           RFC 1034, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", RFC 2119, March 1997.

   [RFC2203] Eisler, M. and J. Linn, "RPCSEC_GSS Protocol
              Specification", RFC 2203, September 1997.

   [RFC2743] Linn, J., "Generic Security Service Application Program
              Interface Version 2, Update 1", RFC 2743, January 2000.

   [RFC4121] Zhu, L., Jaganathan, K., and S. Hartman, "The Kerberos
              Version 5 Generic Security Service Application Program
              Interface (GSS-API) Mechanism: Version 2", RFC 4121, July
              2005.

   [RFC4511] Sermersheim, Ed., J., "Lightweight Directory Access
              Protocol (LDAP): The Protocol", RFC 4511, June 2006.

   [RFC5661] Shepler, S., Eisler, M., and D. Noveck, "Network File
              System (NFS) Version 4 Minor Version 1 Protocol", RFC
              5661, January 2010.

   [RFC7530] Haynes, T. and D. Noveck, "Network File System (NFS)
              version 4 Protocol", RFC 7530, March 2015.

10.2. Informative References

   [CIFS]   Microsoft Corporation, "[MS-CIFS] -- v20130118 Common
              Internet File System (CIFS) Protocol", January 2013.

   [I-D.sorce-krbwg-general-pac]
              Sorce, S., Yu, T., and T. Hardjono, "A Generalized PAC for
              Kerberos V5", draft-ietf-krb-wg-general-pac-01 (Work In
              Progress awaiting merge with other document), June 2011.

   [PAC]   Brezak, J., "Utilizing the Windows 2000 Authorization Data
              in Kerberos Tickets for Access Control to Resources",
              October 2002.

   [RFC2307] Howard, L., "An Approach for Using LDAP as a Network
              Information Service", RFC 2307, March 1998.

   [RFC5716] Lentini, J., Everhart, C., Ellard, D., Tewari, R., and M.
              Naik, "Requirements for Federated File Systems", RFC 5716,
              January 2010.

Appendix A. Acknowledgments

We thank Chuck Lever, Tom Haynes, Brian Reitz, Bruce Fields, and
David Noveck for their review.

Authors' Addresses

   William A. (Andy) Adamson
   NetApp

   Email: andros@netapp.com

   Nicolas Williams
   Cryptonector

   Email: nico@cryptonector.com