Internet Engineering Task Force (IETF)
Request for Comments: 8000
Category: Standards Track
ISSN: 2070-1721

W. Adamson
NetApp
N. Williams
Cryptonector

November 2016

Requirements for NFSv4 Multi-Domain Namespace Deployment

Abstract

This document presents requirements for the deployment of the NFSv4 protocols for the construction of an NFSv4 file namespace in environments with multiple NFSv4 Domains. To participate in an NFSv4 multi-domain file namespace, the server must offer a multi-domain-capable file system and support RPCSEC_GSS for user authentication. In most instances, the server must also support identity-mapping services.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8000.

Copyright Notice

Table of Contents

1. Introduction

The NFSv4 protocols NFSv4.0 [RFC7530], NFSv4.1 [RFC5661], and NFSv4.2 [RFC7862] introduce the concept of an NFS Domain. An NFSv4 Domain is defined as a set of users and groups using the NFSv4 name@domain user and group identification syntax with the same specified @domain.

Previous versions of the NFS protocol, such as NFSv3 [RFC1813], use the UNIX-centric user identification mechanism of numeric user and group ID for the uid3 and gid3 [RFC1813] file attributes and for identity in the authsys_parms AUTH_SYS credential defined in the Open Network Computing (ONC) Remote Procedure Call (RPC) protocol [RFC5531]. Section 6.1 of [RFC2624] notes that the use of UNIX-centric numeric IDs limits the scale of NFS to large local work groups. UNIX-centric numeric IDs are not unique across NFSv3 deployments and so are not designed for Internet scaling achieved by taking into account multiple naming domains and multiple naming mechanisms (see Section 6.2). The NFSv4 Domain's use of the name@domain syntax provides this Internet scaling by allowing servers and clients to translate between the external name@domain string representation to a local or internal numeric (or other identifier) representation, which matches internal implementation needs.

Multi-domain deployments require support for unique identities across the deployment's name services and security services, as well as the use of multi-domain file systems capable of the on-disk representation of identities belonging to multiple NFSv4 Domains. The name@domain syntax can provide unique identities and thus enables the NFSv4 multi-domain file namespace.

Unlike previous versions of NFS, the NFSv4 protocols define a referral mechanism (Section 8.4.3 of [RFC7530]) that allows a single server or a set of servers to present a multi-server namespace that encompasses file systems located on multiple servers. This enables the establishment of site-wide, organization-wide, or even a truly global file namespace.

The NFSv4 protocols' name@domain syntax and referral mechanism along with the use of RPCSEC_GSS security mechanisms enables the construction of an NFSv4 multi-domain file namespace.

This document presents requirements on the deployment of the NFSv4 protocols for the construction of an NFSv4 file namespace in environments with multiple NFSv4 Domains. To participate in an NFSv4 multi-domain file namespace, the server must offer a multi-domain-capable file system and support RPCSEC_GSS [RFC2203] for user authentication. In most instances, the server must also support identity-mapping services.

1.1. 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].

2. Terminology

NFSv4 Domain: A set of users and groups using the NFSv4 name@domain user and group identification syntax with the same specified @domain.

Stand-alone NFSv4 Domain: A deployment of the NFSv4 protocols and NFSv4 file namespace in an environment with a single NFSv4 Domain.

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 APIs. Examples are numeric IDs such as a uidNumber (UID), gidNumber (GID) [RFC2307], or a Windows Security Identifier (SID) [CIFS]. In some cases, 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.

Unique identity: An on-the-wire form of identity that is unique across an NFSv4 multi-domain namespace that can be mapped to a local representation. For example, the NFSv4 name@domain or the Kerberos principal [RFC4120].

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

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

Principal: An RPCSEC_GSS [RFC2203] authentication identity. It is usually, but not always, a user; rarely, if ever, a group; and sometimes a host or server.

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

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

Name Service: Facilities that provide 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 unique identities or principals from within local and remote domains. Often provided by a Directory Service such as the Lightweight Directory Access Protocol (LDAP) [RFC4511].

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

FedFS: 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.

Domain: This term is used in multiple contexts where it has different meanings. "NFSv4 Domain" and "multi-domain" are defined above.

DNS domain: A set of computers, services, or any Internet resource identified by a 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.

3. Federated File System

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

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 5) are not compatible with joining a multi-domain FedFS namespace.

4. Identity Mapping

4.1. 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 may be required to perform two kinds of mappings depending upon what authentication and authorization information is sent on the wire and what is stored in the exported file system. For example, if an authentication identity such as a Kerberos principal is sent with authorization information such as a "privilege attribute certificate" (PAC) [PAC], then mapping is not required (see Section 8).

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. Metadata 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.

4.2. NFSv4 Client Identity Mapping

A client setting the owner or group attribute will often need access to identity-mapping services. This is because APIs 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 unique identity 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 unique identity to a local form.

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

5. Stand-Alone NFSv4 Domain Deployment Examples

The purpose of this section is to list some typical stand-alone deployment examples to highlight the need for the required restraints to the NFSv4 protocol, name service configuration, and security service choices in an NFSv4 multi-domain environment described in Section 6.

Section 7 notes how these stand-alone deployment examples would need to change to participate in an NFSv4 multi-domain deployment.

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.

These examples are for an 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.

5.1. AUTH_SYS with Stringified UID/GID

This example is the closest NFSv4 gets to being run as NFSv3 as there is no need for a name service for file metadata listing.

File access: The AUTH_SYS RPC credential [RFC5531] 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.

Metadata 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.

5.2. 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 5.1.

Metadata 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 not an issue in a stand-alone NFSv4 Domain deployment as the NFSv4 Domain is known to the NFSv4 server and can be combined with the login name to form the name@domain syntax after the return of the name service call.

5.3. RPCSEC_GSS with Name@domain

RPCSEC_GSS uses Generic Security Service Application Program Interface (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.

Metadata setting and listing: This is the same as in Section 5.2.

6. Multi-Domain Constraints to the NFSv4 Protocol

Joining NFSv4 Domains under a single file namespace imposes slightly on the NFSv4 administrative freedom. In this section, we describe the required constraints.

6.1. Name@domain Constraints

NFSv4 uses a syntax of the form "name@domain" (see Section 5.9 of [RFC7530]) 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.

- 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.

- 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.

6.1.1. 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, the @domain portion of the name@domain syntax, needs clarification to work in a multi-domain file system namespace. [RFC5661], Section 5.9 loosely defines the NFSv4 Domain name as a DNS domain name. This loose definition for the NFSv4 Domain name is a good one, as DNS domain names are globally unique. As noted in Section 6.1, any choice of NFSv4 Domain name can work within a stand-alone NFSv4 Domain deployment whereas the NFSv4 Domain name is required to be unique across 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.

6.1.2. NFSv4 Domain and Name Services

As noted in Section 6.1, 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.

6.2. 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 GSS-API [RFC2203] 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 service.

The AUTH_NONE [RFC5531] security flavor can be useful in a multi-domain deployment to grant universal read-only access to public data without any credentials.

The AUTH_SYS security flavor [RFC5531] 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 a name service as described in [RFC2307]. 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.

6.2.1. NFSv4 Domain and Security Services

As noted in Section 6.2 regarding AUTH_NONE, 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 need of mapping multiple Kerberos 5 principal@REALMs to the same local ID. Methods of achieving this are beyond the scope of this document.

7. Stand-Alone Examples in an NFSv4 Multi-Domain Deployment

In this section, we revisit the stand-alone NFSv4 Domain deployment examples in Section 5 and note what is prohibiting them from participating in an NFSv4 multi-domain deployment.

Note that 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. To participate in an NFSv4 multi-domain deployment, all three examples in Section 5 would need to export multi-domain-capable file systems.

Due to the use of AUTH_SYS and stringified UID/GIDs, the first stand-alone deployment example (described in Section 5.1) is not suitable for participation in an NFSv4 multi-domain deployment.

The second example (described in Section 5.2) does use the name@domain syntax, but the use of AUTH_SYS prohibits its participation in an NFSv4 multi-domain deployment.

The third example (described in Section 5.3) can participate in a multi-domain namespace deployment if:

- The NFSv4 Domain name is unique across the namespace.

- All exported file systems are multi-domain capable.

- A secure method is used to resolve the remote NFSv4 Domain principal's authorization information from an authoritative source.

8. 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 SHOULD 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. For example, 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 principal's NFSv4 Domain. In this case, there is no assumption of:

- Remote name service configuration knowledge.

- 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 detailing these methods 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 "principal authorization data" (PAD) [GEN-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 has a well-known syntax.

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 principals in the remote NFSv4 Domain. In this case, the NFSv4 server makes a query to its local NFSv4 Domain name service just as it does when servicing a local domain principal. While this requires detailed knowledge of the remote NFSv4 Domain 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.

9. 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], Requirements for Federated FS [RFC5716], FedFS Namespace Database Protocol [RFC7532], FedFS Administration Protocol [RFC7533], and FedFS Security Addendum [SEC-ADD] apply.

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

- latency in propagation of revocation of authentication credentials
- latency in propagation of revocation of authorizations
- latency in propagation of granting of authorizations
- complications in establishing a complete authorization context for users of a foreign domain (only parts may be available to servers)

- 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.

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 (see Section 8). In the simplest scheme, a server has access to a database mapping all known principal names to user names whose authorization context can be evaluated using operating system interfaces that deal in user names rather than principal names.

Note that clients may also need to evaluate a server’s authorization context when using labeled security [RFC7862] (e.g., is the server authorized to handle content at a given security level for the given client process subject label).

When the server accepts user credentials from more than one realm, it is important to remember that the server must verify that the client it is talking to has a credential for the name the client has presented the server and that the credential’s issuer (i.e., its realm) is allowed to issue it. Usually, the service principal realm authorization function is implemented by the security mechanism, but the implementor should check this.

10. References

10.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC1813] Callaghan, B., Pawlowski, B., and P. Staubach, "NFS Version 3 Protocol Specification", RFC 1813, DOI 10.17487/RFC1813, June 1995, <http://www.rfc-editor.org/info/rfc1813>.

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

[RFC2203] Eisler, M., Chiu, A., and L. Ling, "RPCSEC_GSS Protocol Specification", RFC 2203, DOI 10.17487/RFC2203, September 1997, <http://www.rfc-editor.org/info/rfc2203>.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, DOI 10.17487/RFC2743, January 2000, <http://www.rfc-editor.org/info/rfc2743>.

[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, DOI 10.17487/RFC4121, July 2005, <http://www.rfc-editor.org/info/rfc4121>.

[RFC4511] Sermersheim, J., Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, DOI 10.17487/RFC4511, June 2006, <http://www.rfc-editor.org/info/rfc4511>.

[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, <http://www.rfc-editor.org/info/rfc5661>.

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

[RFC7862] Haynes, T., "Network File System (NFS) Version 4 Minor Version 2 Protocol", RFC 7862, DOI 10.17487/RFC7862, November 2016, <http://www.rfc-editor.org/info/rfc7862>.

10.2. Informative References

[CIFS] Microsoft Corporation, "[MS-CIFS]: Common Internet File System (CIFS) Protocol", MS-CIFS v20160714 (Rev 26.0), July 2016.

[GEN-PAC] Sorce, S., Ed., Yu, T., Ed., and T. Hardjono, Ed., "A Generalized PAC for Kerberos V5", Work in Progress, draft-ietf-krb-wg-general-pac-01, October 2011.

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

[RFC2307] Howard, L., "An Approach for Using LDAP as a Network Information Service", RFC 2307, DOI 10.17487/RFC2307, March 1998, <http://www.rfc-editor.org/info/rfc2307>.

[RFC2624] Shepler, S., "NFS Version 4 Design Considerations", RFC 2624, DOI 10.17487/RFC2624, June 1999, <http://www.rfc-editor.org/info/rfc2624>.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, DOI 10.17487/RFC4120, July 2005, <http://www.rfc-editor.org/info/rfc4120>.

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

[RFC5716] Lentini, J., Everhart, C., Ellard, D., Tewari, R., and M. Naik, "Requirements for Federated File Systems", RFC 5716, DOI 10.17487/RFC5716, January 2010, <http://www.rfc-editor.org/info/rfc5716>.

[RFC7532] Lentini, J., Tewari, R., and C. Lever, Ed., "Namespace Database (NSDB) Protocol for Federated File Systems", RFC 7532, DOI 10.17487/RFC7532, March 2015, <http://www.rfc-editor.org/info/rfc7532>.

[RFC7533] Lentini, J., Tewari, R., and C. Lever, Ed., "Administration Protocol for Federated File Systems", RFC 7533, DOI 10.17487/RFC7533, March 2015, <http://www.rfc-editor.org/info/rfc7533>.

[SEC-ADD] Lever, C., "Federated Filesystem Security Addendum", Work in Progress, draft-cel-nfsv4-federated-fs-security-addendum-06, October 2016.

Acknowledgments

Andy Adamson would like to thank NetApp, Inc., for its funding of his time on this project.

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