Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 27, 2011

C. Everhart
W. Adamson
NetApp
J. Zhang
Google

February 23, 2011

Using DNS SRV to Specify a Global File Name Space with NFS version 4
draft-ietf-nfsv4-federated-fs-dns-srv-namespace-07.txt

Abstract

The NFS version 4 protocol provides a natural way for a collection of NFS file servers to collaborate in providing an organization-wide file name space. The DNS SRV RR allows a simple and appropriate way for an organization to publish the root of its name space, even to clients that might not be intimately associated with such an organization. The DNS SRV RR can be used to join these organization-wide file name spaces together to allow construction of a global, uniform NFS version 4 file name space.

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 http://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 27, 2011.

Table of Contents

1. Requirements notation

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. Background

The NFS Version 4 protocol [RFC3530] introduced the fs_locations attribute. Its use was elaborated further in the NFS Version 4 Minor Version 1 protocol [RFC5661], which also defined an extended version of the attribute as fs_locations_info. With the advent of these attributes, NFS servers can cooperate to build a file name space that crosses server boundaries. The fs_locations and fs_locations_info attributes are used as referrals, so that a file server may indicate to its client that the file name tree beneath a given name in the server is not present on itself, but is represented by a filesystem in some other set of servers. The mechanism is general, allowing servers to describe any filesystem as being reachable by requests to any of a set of servers. Thus, starting with a single NFS Version 4 server, using these referrals, an NFS Version 4 client might be able to see a large name space associated with a collection of interrelated NFS Version 4 file servers. An organization could use this capability to construct a uniform file name space for itself.

An organization might wish to publish the starting point for this name space to its clients. In many cases, the organization will want to publish this starting point to a broader set of possible clients. At the same time, it is useful to require clients to know only the smallest amount of information in order to locate the appropriate name space. Simultaneously, that required information should be constant through the life of an organization if the clients are not to require reconfiguration as administrative events change, for instance, a server's name or address.

3. Proposed Use of SRV Resource Record in DNS

Providing an organization's published filesystem name space is a service, and it is appropriate to use the DNS [RFC1034][RFC1035] to locate it. As with the AFSDB resource record type [RFC1183], the client need only utter the (relatively) constant domain name for an organization in order to locate its filesystem name space service. Once a client uses the DNS to locate one or more servers for the root of the organization's name space, it can use the standard NFS Version 4 mechanisms to navigate the remainder of the NFS servers for that organization. The use of this proposed mechanism results in a useful

cross-organizational name space, just as in AFS [AFS] and DCE/DFS [DFS] before it. A client need know only the name of the organization in order to locate the filesystem name space published by that organization.

We propose the use of the DNS SRV resource record type [RFC2782] to fulfill this function. The format of the DNS SRV record is as follows:

```
_Service._Proto.Name TTL Class SRV Priority Weight Port Target
```

In our case, we use a Service name of "_nfs4._domainroot" and a conventional Protocol of "_tcp". The Target fields give the domain names of the NFS Version 4 servers that export filesystems for the domain's root. An NFS Version 4 client SHOULD interpret any of the exported root filesystems as the filesystem published by the organization with the given domain name.

In order to allow the NFSv4 servers so given to export a variety of filesystems, those file servers SHOULD export the given domain's root filesystems at "/.domainroot-{Name}" within their pseudo-filesystems, where the "{Name}" is the name of the organization as used in the SRV RR.

As an example, suppose a client wished to locate the root of the filesystem published by organization example.net. The DNS servers for the domain could publish records like

```
$ORIGIN example.net.
_nfs4._domainroot._tcp IN SRV 0 0 2049 nfs1tr.example.net.
_nfs4._domainroot._tcp IN SRV 1 0 2049 nfs2ex.example.net.
```

The result domain names nfs1tr.example.net and nfs2ex.example.net indicate NFS Version 4 file servers that export the root of the published name space for the example.net domain. In accordance with RFC 2782 [RFC2782], these records are to be interpreted using the Priority and Weight field values, selecting an appropriate file server with which to begin a network conversation. The two file servers would export filesystems that would be found at "/.domainroot-example.net" in their pseudo-filesystems, which clients would mount. Clients then carry out subsequent accesses in accordance with the ordinary NFS Version 4 protocol.

We use a composite Service name (built from "_nfs4" and "_domainroot") so that other filesystem protocols could make use of the same "_domainroot" abstraction.

4. Integration with Use of NFS Version 4

We expect that NFSv4 clients will implement a special directory, analogous to an Automounter [AMD] directory, the entries in which are domain names that have recently been traversed. When an application attempts to traverse a new name in that special directory, the NFSv4 client consults DNS to obtain the SRV data for the given name, and if successful, it mounts the indicated filesystem(s) in that name in the special directory. The goal is that NFSv4 applications will be able to lookup an organization's domain name in the special directory, and the NFSv4 client will be able to discover the filesystem that that organization publishes. Entries in the special directory will be domain names, and they will each appear to the application as a directory name pointing to the root directory of the filesystem published by the organization responsible for that domain name.

This functionality does not require or use any list of organizations that are known to provide file service, as AFS did with its "root.afs" functionality.

This DNS SRV record evaluation could, in principle, be done either in the NFSv4 client or in an NFSv4 server. In either case, the result would appear the same to applications on the NFSv4 client.

4.1. Globally-useful names: conventional mount point

For the inter-organizational name space to be a global name space, it is useful for its mount point in local systems to be uniform as well. On POSIX machines, the name /nfs4/ SHOULD be used so that names on one machine will< be directly usable on any machine. Thus, the example.net published filesystem would be accessible as

```
/nfs4/example.net/
```

on any POSIX client. Using this convention, "/nfs4/" is the name of the special directory that is populated with domain names, leading to file servers and filesystems that capture the results of SRV record lookups.

4.2. Mount options

SRV records are necessarily less complete than the information in the existing NFS Version 4 attributes fs_locations [RFC3530] or fs_locations_info [RFC5661]. For the rootpath field of fs_location, or the fli_fs_root of fs_locations_info, we use the "/.domainroot-{Name}" string. Thus, the servers listed as targets for the SRV resource records should export the root of the organization's published filesystem as the directory "/.domainroot-{Name}" (for the

given organization Name) in its exported namespace. For example, for organization "example.net", the directory "/.domainroot-example.net" should be used.

As for the other attributes in fs_locations_info, the recommended approach is for a client to make its first possible contact with any of the referred-to servers, obtain the fs_locations_info structure from that server, and use the information from that obtained structure as the basis for its judgment of whether it would be better to use a different server representative from the set of servers for that filesystem.

The process of mounting an organization's name space should permit the use of what is likely to impose the lowest cost on the server. Thus, the NFS client SHOULD NOT insist on using a writable copy of the filesystem if read-only copies exist, or a zero-age copy rather than a copy that may be a little older. We presume that the organization's file name space can be navigated to provide access to higher-cost properties such as writability or currency as necessary, but that the default use when navigating to the base information for an organization ought to be as low-overhead as possible.

Because of the possible distinction between read-only and read-write versions of a filesystem, organizations SHOULD also publish the location of a writable instance of their root filesystems, and that NFSv4 clients SHOULD mount that filesystem under the organizational domain name preceded by a period ("."). Therefore, when names beginning with a period are looked up under the NFSv4 client's special directory, the SRV RR looked up in DNS uses a Service name of "_nfs4._write._domainroot", and the indicated server (or servers) SHOULD export the writable instance at "/.domainroot-write-{Name}" for a domain name Name.

Extending the opening example, suppose a client wished to locate the read-only and read-write roots of the filesystem published by organization example.net. Suppose a read-write instance were hosted on server nfs1tr.example.net, and read-only instances were on that server and also on server nfs2ex.example.net. The DNS servers for the domain would publish records like

```
$ORIGIN example.net.
_nfs4._domainroot._tcp IN SRV 0 0 2049 nfs1tr.example.net.
_nfs4._domainroot._tcp IN SRV 1 0 2049 nfs2ex.example.net.
_nfs4._write._domainroot._tcp IN SRV 0 0 2049 nfs1tr.example.net.
```

The nfs1tr.example.net server would export filesystems at both "/.domainroot-example.net" (the read-only instance) and "/.domainroot-write-example.net" (the read-write instance). The

nfs2ex.example.net server need export only the "/.domainroot-example.net" name for its read-only instance.

The read-write version of the filesystem would be mounted (upon use) under ".example.net" in the special directory, and a read-only version would be mounted under "example.net". Thus,

```
/nfs4/example.net/users
```

might be a directory in a read-only instance of the root filesystem of the organization "example.net", while

```
/nfs4/.example.net/users
```

would be a writable form of that same directory. A small benefit of following this convention is that names with the period prefix are treated as "hidden" in many operating systems, so that the visible name remains the lowest-overhead name.

4.3. Filesystem integration issues

The result of the DNS search SHOULD appear as a (pseudo-)directory in the client name space. A further refinement is advisable, and SHOULD be deployed: that only fully-qualified domain names appear as directories. That is, in many environments, DNS names may be abbreviated from their fully-qualified form. In such circumstances, multiple names might be given to filesystem code that all resolve to the same DNS SRV RRs. The abbreviated form SHOULD be represented in the client's name space cache as a symbolic link, pointing to the fully-qualified name, case-canonicalized when appropriate. This will allow pathnames obtained with, say, getcwd() to include the DNS name that is most likely to be usable outside the scope of any particular DNS abbreviation convention.

5. Where is this integration carried out?

Another consideration is what agent should be responsible for interpreting the SRV records. It could be done just as well by the NFS client or by the NFS server, though we expect that most clients will include this function themselves. Using something like Automounter [AMD] technology, the client would be responsible for interpreting names under a particular directory, discovering the appropriate filesystem to mount, and mounting it in the appropriate place in the client name space before returning control to the application doing a lookup. Alternatively, one could imagine the existence of an NFS version 4 server that awaited similar domain-name lookups, then consulted the SRV records in DNS to determine the

servers for the indicated published filesystem, and then returned that information as an NFS Version 4 referral. In either case, the result of the DNS lookup should be cached (obeying TTL) so that the result could be returned more quickly the next time.

We strongly suggest that this functionality be implemented by NFS clients. While we recognize that it would be possible to configure clients so that they relied on a specially-configured server to do their SRV lookups for them, we feel that such a requirement would impose unusual dependencies and vulnerabilities for the deployers of such clients. Yet even if it is desirable to deploy this functionality on the NFS client side, it may be valuable as a transition aid for a site to be able to deploy it on the NFS server side: it may be easier for them to install it on special NFS servers rather than install it on all their NFS clients. Thus, from an implementation standpoint, NFS clients SHOULD implement the functionality, and NFS servers MAY implement it.

6. Security Considerations

Naive use of the DNS may effectively give clients published server referrals that are intrusive substitutes for the servers intended by domain administrators.

It may be possible to build a trust chain by using DNSSEC [RFC4033] to implement this function on the client, or by implementing this function on an NFS Version 4 server that uses DNSSEC and maintaining a trust relationship with that server. This trust chain also breaks if the SRV interpreter accepts responses from insecure DNS zones. Thus, it would likely be prudent also to use domain-based service principal names for the servers for the root filesystems as indicated as the targets of the SRV records. The idea here is that one wants to authenticate {nfs, domainname, host.fqdn}, not simply {nfs, host.fqdn}, when the server is a domain's root file server obtained through an insecure DNS SRV RR lookup. The domain administrator can thus ensure that only domain root NFSv4 servers have credentials for such domain-based service principal names.

Domain-based service principal names are defined in RFCs 5178 [RFC5178] and 5179 [RFC5179]. To make use of RFC 5178's domain-based names, the syntax for "domain-based-name" MUST be used with a service of "nfs", a domain matching the name of the organization whose root filesystem is being sought, and a hostname given in the target of the DNS SRV resource record. Thus, in the example above, two file servers (nfs1tr.example.net and nfs2ex.example.net) are located as hosting the root filesystem for the organization example.net. To communicate with, for instance, the second of the given file servers,

GSS-API should be used with the name-type of GSS_C_NT_DOMAINBASED_SERVICE defined in RFC 5178 and with a symbolic name of

nfs@example.net@nfs2ex.example.net

in order to verify that the named server (nfs2ex.example.net) is authorized to provide the root filesystem for the example.net organization.

7. IANA Considerations

None.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", RFC 1035, November 1987.

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

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3530] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M., and D. Noveck, "Network File System (NFS) version 4 Protocol", RFC 3530, April 2003.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC5178] Williams, N. and A. Melnikov, "Generic Security Service Application Program Interface (GSS-API) Internationalization and Domain-Based Service Names and Name Type", RFC 5178, May 2008.

[RFC5179] Williams, N., "Generic Security Service Application Program Interface (GSS-API) Domain-Based Service Names

Mapping for the Kerberos V GSS Mechanism", RFC 5179, May 2008.

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

8.2. Informative References

[AFS] Howard, J., "An Overview of the Andrew File System"", Proc. USENIX Winter Tech. Conf. Dallas, February 1988.

[AMD] Pendry, J. and N. Williams, "Amd: The 4.4 BSD Automounter Reference Manual", March 1991, <http://docs.freebsd.org/info/amdref/amdref.pdf>.

[DFS] Kazar, M., Leverett, B., Anderson, O., Apostolides, V., Bottos, B., Chutani, S., Everhart, C., Mason, W., Tu, S., and E. Zayas, "DEcorum File System Architectural Overview", Proc. USENIX Summer Conf. Anaheim, Calif., June 1990.

[RFC1183] Everhart, C., Mamakos, L., Ullmann, R., and P. Mockapetris, "New DNS RR Definitions", RFC 1183, October 1990.

Authors' Addresses

Craig Everhart
NetApp
800 Cranberry Woods Drive, Ste. 300
Cranberry Township, PA 16066
US

Phone: +1 724 741 5101
Email: everhart@netapp.com

W.A. (Andy) Adamson
NetApp
495 East Java Drive
Sunnyvale, CA 94089
US

Phone: +1 734 665 1204
Email: andros@netapp.com

Jiaying Zhang
Google
604 Arizona Avenue
Santa Monica, CA 90401
US

Phone: +1 310 309 6884
Email: jiayingz@google.com