Requirments for the Nameserver Communication protocol
draft-regnauld-ns-communication-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on June 22, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).
Abstract

This document describes the requirements for a protocol to allow DNS nameservers to communicate among themselves, possibly outside the existing DNS protocol, for purposes of zone discovery and provisioning and remote management.

Table of Contents

1. Requirements notation ........................................ 3
2. Introduction .................................................... 4
3. Use cases ........................................................ 5
   3.1. Having a homogeneous view of non-standard zones ..... 5
   3.2. Exchanging secondary name service with partners ..... 5
   3.3. Managing remote name servers ......................... 5
4. Terminology ..................................................... 7
5. The requirements ................................................ 8
6. Security Considerations ....................................... 10
7. References ...................................................... 11
   7.1. Normative References .................................. 11
   7.2. Informative References ................................ 11
Appendix A. Related work ....................................... 12
Appendix B. Acknowledgements .................................. 13
Authors’ Addresses ............................................... 14
Intellectual Property and Copyright Statements ............... 15
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 [3].

As in other RFC describing requirements (such as [5]), the MUST and MUST NOT have to be interpreted in terms of the protocol, not in terms of the implementation. If this document says that the protocol MUST do something, it means that the protocol must have a way to describe this "something", not that every implementation has to implement it or that every running instance has to allow it. For instance, the AXFR zone transfer of the DNS is a MUST of the protocol but an implementation may omit it and a specific nameserver is certainly free to disable it.
2. Introduction

Administrators of DNS ([1], [2]) Nameservers often need to communicate to exchange meta-information, which cannot be distributed by the DNS, or to request various administrative tasks such as the provisioning of a new zone or querying the list of managed zones. Currently, this is done by ad hoc means, often manually. When a formal protocol exists (see Appendix A), it is always proprietary and undocumented. We believe it would be nice to have a "nameserver communication" protocol to address these needs.

This document specifies the requirements for such a protocol. The choice is to define one protocol for all the currently out-of-band tasks of nameserver management. The rationale is that each task is quite simple and relatively easy to define so one protocol for each task would be overkill.

The use cases below describe typical scenarios where such a protocol would be useful for the nameserver administrators.
3. Use cases

3.1. Having a homogeneous view of non-standard zones

If an organization uses non-standard zones (such as a purely-local TLD, but not only), synchronizing all the nameservers so they all see these zones is usually a time-consuming task. It is even worse if two such organizations merge.

This is typically done, when using BIND ([7]), with "stub" or "forward" zones. But there is no way to ensure automatically that all the resolvers have the same set of zones at a given time, as new zones may be added locally, without the other nameservers being updated to reflect the change. The problem can be mitigated by concentrating DNS queries through a core set of nameservers, but these still require to be updated when zones are added on the various nameservers.

Though the architecture described above is not recommended, such scenarios do happen in the wild, especially when several large organizations are connected together.

3.2. Exchanging secondary name service with partners

A large ISP may manage thousands of zones. For reliability reasons, following [4], it should have secondary nameservers placed in widely different geographic areas. A common solution is to establish cross-hosting agreements (zone exchange) with a partner in the "I’ll host your zones and you’ll host mine" fashion.

In the absence of a standard protocol to discover the available zones in the scope of this agreement (one server may be secondary for several other organizations), such exchange of service has to be done manually, each time a zone is created or deleted, or through proprietary means such as email in a structured, authenticated format.

The above is equally true for the management of nameservers within a single organization, in order to provision secondary nameservers located in affiliate / branch offices from a central location in an automatic fashion.

3.3. Managing remote name servers

To implement the recommendations of [4], a one-site organization typically needs to request secondary hosting located on a distinct organization’s premises.
Many ccTLDs do so, for instance. But, in that case, the "outsourcing" organization has typically no control of the nameservers. Changing the IP address of the master nameserver, or forcing a reload when a SOA serial number accidentally wrapped is a manual operation at the remote site, and may require the intervention, and availability, of the staff at the remote organization.
4. Terminology

Since the two participants in a conversation are servers, we use the following vocabulary:

- **Requestor**: the server which requests something (an information or an action),

- **Responder**: the server which will reply and may be act.
5. The requirements

The protocol:

1. MUST allow the requestor to authenticate the responder and vice-versa.

2. MUST allow the communication to be hidden from snoopers (relying on a transport like TLS - [6] - is possible).

3. MUST provide anti-replay protection (making all the requests idempotent is a possible way),

4. MUST provide a standard vocabulary to express the types of zones ("master" or "slave" but also the non-standard terms "stub" and "forward" in BIND, zones for which the responder is authoritative but does not have the data itself).

5. MUST honor existing definition of views based on predefined shared key or IP address scope, so that information specific to a particular view, and only this information, will be returned to the requestor if the requestor’s credentials match those of the defined view (for example, a zone may be of type master in one view, and forward in another). DISCUSSION: a requestor can manage views where its IP address would show it another view. May be explicitly naming views would be better?

6. MUST provide a way to query a responder on the zones it serves with authority. The response format MUST allow to carry other information besides the zone name, such as the type of the zone.

7. MUST provide a way to request the provisioning of a new zone. It MUST allow for at least the following parameters:
   1. IP address(es) of the master(s),
   2. type of the zone,
   3. and may be contact information?

8. MUST allow a way to tell the responder, if it is acting as a slave, to reload a zone, regardless of the current value of the SOA serial number.

9. SHOULD be implementable as a one request / one response system where the request is self-sufficient and carries all the information that the responder needs. Such a system would allow for various transports such as email or simple TCP mapping.
10. MAY provide a standard vocabulary to express protections (ACL) for a zone. DISCUSSION: it would be a very good thing but standardizing ACL language may be a daunting task. A simpler solution may be simply to have a boolean telling if the zone is private (internal to an organization) or public. Additionnally, it may be required to implement a concept of scope, so that it can be specified which agreement the zone is bound to (in the case of many-to-many secondary cross-hosting relationships). (Another way to implement scope could be to use the identity of the requestor, since it is authenticated. See the next requirement.)
6. Security Considerations

Allowing remote configuration of a nameserver is a very sensitive issue, particularly when said server may be serving other third-party zones. Therefore the security requirements listed above (allowing reciprocal authentication and protection against snoopers) are a MUST.

There are no other requirements for the protocol itself but, for the implementations, it is reasonable to ask that, by default, all requests via this protocol must be denied.

In the same way, authorisation of requestors, once they are authenticated by the protocol, is up to the implementations. They should allow fine-grained configuration of permissions. Implementors should be warned that the authorized requestors of a responder may not trust each other.
7. References

7.1. Normative References


7.2. Informative References


Appendix A. Related work

PowerDNS ([8]) has a proprietary protocol, Supermaster, which allows remote provisioning of zones: See

Microsoft AD/DNS and Infoblox also have such a protocol: a zone created on the master can be created automatically on the slaves.
Appendix B. Acknowledgements

The original idea came from Bert Hubert. Edward Lewis, by its nice explanations on the bind-users mailing list, helped a lot. Useful comments from Mohsen Souissi were integrated.
Authors’ Addresses

Phil Regnauld
catpipe ApS
Snaregade 12, 2tv
Copenhagen K 1205
Denmark

Email: regnauld@catpipe.net

Stephane Bortzmeyer
AFNIC
Immeuble International
Saint-Quentin-en-Yvelines 78181
France

Phone: +33 1 39 30 83 46
Email: bortzmeyer+ietf@nic.fr
URI:  http://www.afnic.fr/