Internet Engineering Task Force (IETF)
Request for Comments: 6168
Category: Informational
ISSN: 2070-1721

W. Hardaker
Sparta, Inc.

May 2011

# Requirements for Management of Name Servers for the DNS

## Abstract

Management of name servers for the Domain Name System (DNS) has traditionally been done using vendor-specific monitoring, configuration, and control methods. Although some service monitoring platforms can test the functionality of the DNS itself, there is not an interoperable way to manage (monitor, control, and configure) the internal aspects of a name server itself.

This document discusses the requirements of a management system for name servers and can be used as a shopping list of needed features for such a system.

## Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

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/rfc6168.

Table of Contents

## 1. Introduction

Management of name servers for the Domain Name System (DNS) [RFC1034] [RFC1035] has traditionally been done using vendor-specific monitoring, configuration, and control methods. Although some service monitoring platforms can test the functionality of the DNS itself, there is not an interoperable way to manage (monitor, control, and configure) the internal aspects of a name server itself.

Previous standardization work within the IETF resulted in the creation of two SNMP MIB modules [RFC1611] [RFC1612], but they failed to achieve significant implementation and deployment. The perceived reasons behind the failure for the two MIB modules are documented in [RFC3197].

This document discusses the requirements of a management system for name servers and can be used as a shopping list of needed features for such a system. This document only discusses requirements for managing the name server component of a system -- not other elements of the system itself.

Specifically out of scope for this document are requirements associated with the management of stub resolvers. It is not the intent of this document to document stub resolver requirements, although some of the requirements listed are applicable to stub resolvers as well.

The task of creating a management system for managing DNS servers is not expected to be a small one. It is likely that components of the solution will need to be designed in parts over time; these requirements take this into consideration. In particular, Section 5.1 discusses the need for future extensibility of the base management solution. This document is intended to be a roadmap towards a desired outcome and is not intended to define an "all-or-nothing" system. Successful interoperable management of name servers, even in part, is expected to be beneficial to network operators compared to the entirely custom solutions that are used at the time of this writing.

### 1.1. 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].

### 1.2. Terminology

This document is consistent with the terminology defined in Section 2 of [RFC4033]. Additional terminology needed for this document is described below:

Name Server: When we are discussing servers that don't fall into a more specific type of server category defined in other documents, this document will refer to them generically as "name servers". In particular, "name servers" can be considered to be any valid combination of authoritative, recursive, validating, or security-aware. The more specific name server labels will be used when this document refers only to a specific type of server. However, the term "name server", in this document, will not include stub resolvers.

### 1.3. Document Layout and Requirements

This document is written so that each numbered section will contain only a single requirement if it contains one at all. Each requirement will contain needed wording from the terminology described in Section 1.1. Subsections, however, might exist with additional related requirements. The document is laid out in this way so that a specific requirement can be uniquely referred to using the section number itself and the document version from which it came.

## 2. Management Architecture Requirements

This section discusses requirements that reflect the needs of the expected deployment environments.

### 2.1. Expected Deployment Scenarios

DNS zones vary greatly in the type of content published within them. Name servers, too, are deployed with a wide variety of configurations to support the diversity of the deployed content. It is important that a management solution trying to meet the criteria specified in this document consider supporting the largest number of potential deployment cases as possible. Further deployment scenarios that are not used as direct examples of specific requirements are listed in Appendix A.

#### 2.1.1. Zone Size Constraints

The management solution MUST support both name servers that are serving a small number of potentially very large zones (e.g., Top Level Domains (TLDs)) as well as name servers that are serving a very large number of small zones. Both deployment scenarios are common.

#### 2.1.2. Name Server Discovery

Large enterprise network deployments may contain multiple name servers operating within the boundaries of the enterprise network. These name servers may each be serving multiple zones both in and out of the parent enterprise's zone. Finding and managing large numbers of name servers would be a useful feature of the resulting management solution. The management solution MAY support the ability to discover previously unknown instances of name servers operating within a deployed network.

#### 2.1.3. Configuration Data Volatility

Configuration data is defined as data that relates only to the configuration of a server and the zones it serves. It specifically does not include data from the zone contents that is served through DNS itself. The solution MUST support servers that remain statically configured over time as well as servers that have numerous zones being added and removed within an hour. Both deployment scenarios are common.

#### 2.1.4. Protocol Selection

There are many requirements in this document for many different types of management operations (see Section 3 for further details). It is possible that no one protocol will ideally fill all the needs of the requirements listed in this document, and thus multiple protocols might be needed to produce a completely functional management system. Multiple protocols might be used to create the complete management solution, but the solution SHOULD require only one.

#### 2.1.5. Common Data Model

Defining a standardized protocol (or set of protocols) to use for managing name servers would be a step forward in achieving an interoperable management solution. However, just defining a protocol to use by itself would not achieve the entire end goal of a complete interoperable management solution. Devices also need to represent their internal management interface using a common management data model. The solution MUST create a common data model that management stations can make use of when sending or collecting data from a

managed device so it can successfully manage equipment from vendors as if they were generic DNS servers. This common data model is needed for the operations discussion in Section 3. Note that this does not preclude the fact that name server vendors might provide additional management infrastructure beyond a base management specification, as discussed further in Section 5.1.

#### 2.1.6. Operational Impact

It is impossible to add new features to an existing server (such as the inclusion of a management infrastructure) and not impact the existing service and/or server in some fashion. At a minimum, for example, more memory, disk, and/or CPU resources will be required to implement a new management system. However, the impact to the existing DNS service MUST be minimized since the DNS service itself is still the primary service to be offered by the modified name server. The management solution MUST NOT result in an increase of the number of unhandled DNS requests.

### 2.2. Name Server Types

There are multiple ways in which name servers can be deployed. Name servers can take on any of the following roles:

- Master Servers
- Slave Servers
- Recursive Servers

The management solution SHOULD support all of these types of name servers as they are all equally important. Note that "Recursive Servers" can be further broken down by the security sub-roles they might implement, as defined in section 2 of [RFC4033]. These sub-roles are also important to support within any management solution.

As stated earlier, the management of stub resolvers is considered out of scope for this document.

## 3. Management Operation Types

Management operations can traditionally be broken into four categories:

- Control
- Configuration

- Health and Monitoring
- Alarms and Events

This section discusses detailed requirements for each of these four management categories.

## 3.1. Control Requirements

The management solution MUST be capable of performing basic service control operations.

### 3.1.1. Needed Control Operations

These operations SHOULD include, at a minimum, the following operations:

- Starting the name server
- Reloading the service configuration
- Reloading all of the zone data
- Reloading individual zone data sets
- Restarting the name server
- Stopping the name server

Note that no restriction is placed on how the management system implements these operations. In particular, at least "starting the name server" will require a minimal management system component to exist independently of the name server itself.

### 3.1.2. Asynchronous Status Notifications

Some control operations might take a long time to complete. As an example, some name servers take a long time to perform reloads of large zones. Because of these timing issues, the management solution SHOULD take this into consideration and offer a mechanism to ease the burden associated with awaiting the status of a long-running command. This could, for example, result in the use of asynchronous notifications for returning the status of a long-running task, or it might require the management station to poll for the status of a given task using monitoring commands. These and other potential solutions need to be evaluated carefully to select one that balances the result delivery needs with the perceived implementation costs.

Also, see the related discussion in Section 3.4 on notification messages for supporting delivery of alarm and event messages.

## 3.2. Configuration Requirements

Many features of name servers need to be configured before the server can be considered functional. The management solution MUST be able to provide name servers with configuration data. The most important data to be configured, for example, is the served zone data itself.

### 3.2.1. Served Zone Modification

The ability to add, modify, and delete zones being served by name servers is needed. Although there are already solutions for zone content modification (such as Dynamic DNS (DDNS) [RFC2136] [RFC3007], full zone transfer (AXFR) [RFC5936], and incremental zone transfer (IXFR) [RFC1995]) that might be used as part of the final management solution, the management system SHOULD still be able to create a new zone (with enough minimal data to allow the other mechanisms to function as well) and to delete a zone. This might be, for example, a management operation that allows for the creation of at least the initial SOA (Start of Authority) record for a new zone, since that is the minimum amount of zone data needed for the other operations to function.

### 3.2.2. Trust Anchor Management

The solution SHOULD support the ability to add, modify, and delete trust anchors that are used by DNS Security (DNSSEC) [RFC4033] [RFC4034] [RFC4035] [RFC4509] [RFC5011] [RFC5155]. These trust anchors might be configured using the data from the DNSKEY Resource Records (RRs) themselves or by using Delegation Signer (DS) fingerprints.

### 3.2.3. Security Expectations

DNSSEC validating resolvers need to make policy decisions about the requests being processed. For example, they need to be configured with a list of zones expected to be secured by DNSSEC. The management solution SHOULD be able to add, modify, and delete attributes of DNSSEC security policies.

### 3.2.4. TSIG Key Management

TSIG [RFC2845] allows transaction-level authentication of DNS traffic. The management solution SHOULD be able to add, modify, and delete TSIG keys known to the name server.

#### 3.2.5. DNS Protocol Authorization Management

The management solution SHOULD have the ability to add, modify, and delete authorization settings for the DNS protocols itself. Do not confuse this with the ability to manage the authorization associated with the management protocol itself, which is discussed later in Section 4.4. There are a number of authorization settings that are used by a name server. Example authorization settings that the solution might need to cover are:

- Access to operations on zone data (e.g., DDNS)
- Access to certain zone data from certain sources (e.g., from particular network subnets)
- Access to specific DNS protocol services (e.g., recursive service)

Note: the above list is expected to be used as a collection of examples and is not a complete list of needed authorization protections.

### 3.3. Monitoring Requirements

Monitoring is the process of collecting aspects of the internal state of a name server at a given moment in time. The solution MUST be able to monitor the health of a name server to determine its operational status, load, and other internal attributes. Example parameters that the solution might need to collect and analyze are:

- Number of requests sent, responses sent, number of errors, average response latency, and other performance counters
- Server status (e.g., "serving data", "starting up", "shutting down", etc.)
- Access control violations
- List of zones being served
- Detailed statistics about clients interacting with the name server (e.g., top 10 clients requesting data).

Note: the above list is expected to be used as a collection of examples and is not a complete list of needed monitoring operations. In particular, some monitoring statistics are expected to be computationally or resource expensive and are considered to be "nice to have" as opposed to "necessary to have".

### 3.4. Alarm and Event Requirements

Events occurring at the name server that trigger alarm notifications can quickly inform a management station about critical issues. A management solution SHOULD include support for delivery of alarm conditions.

Example alarm conditions might include:

- The server's status is changing (e.g., it is starting up, reloading configuration, restarting or shutting down).
- A needed resource (e.g., memory or disk space) is exhausted or nearing exhaustion.
- An authorization violation was detected.
- The server has not received any data traffic (e.g., DNS requests or NOTIFYs) recently (aka the "lonely warning"). This condition might indicate a problem with the server's deployment.
- The number of errors has exceeded a configured threshold.

## 4. Security Requirements

The management solution will need to be appropriately secured against attacks on the management infrastructure.

### 4.1. Authentication

The solution MUST support mutual authentication. The management client needs to be assured that the management operations are being transferred to and from the correct name server. The managed name server needs to authenticate the system that is accessing the management infrastructure within itself.

### 4.2. Integrity Protection

Management operations MUST be protected from modification while in transit from the management client to the server.

### 4.3. Confidentiality

The management solution MUST support message confidentiality. The potential transfer of sensitive configuration is expected (such as TSIG keys or security policies). The solution does not, however, necessarily need to provide confidentiality to data that would normally be carried without confidentiality by the DNS system itself.

### 4.4. Authorization

The solution SHOULD provide an authorization model capable of selectively authorizing individual management requests for any management protocols it introduces to the completed system. This authorization differs from the authorization previously discussed in Section 3.2.5 in that this requirement is concerned solely with authorization of the management system itself.

There are a number of authorization settings that might be used by a managed system to determine whether the managing entity has authorization to perform the given management task. Example authorization settings that the solution might need to cover are:

- Access to the configuration that specifies which zones are to be served
- Access to the management system infrastructure
- Access to other control operations
- Access to other configuration operations
- Access to monitoring operations

Note: the above list is expected to be used as a collection of examples and is not a complete list of needed authorization protections.

### 4.5. Solution Impacts on Security

The solution MUST minimize the security risks introduced to the complete name server system. It is impossible to add new infrastructure to a server and not impact the security in some fashion as the introduction of a management protocol alone will provide a new avenue for potential attack. Although the added management benefits will be worth the increased risks, the solution still needs to minimize this impact as much as possible.

## 5. Other Requirements

### 5.1. Extensibility

The management solution is expected to change and expand over time as lessons are learned and new DNS features are deployed. Thus, the solution MUST be flexible and able to accommodate new future management operations. The solution might, for example, make use of protocol versioning or capability description exchanges to ensure

that management stations and name servers that weren't written to the same specification version can still interoperate to the best of their combined ability.

#### 5.1.1. Vendor Extensions

It MUST be possible for vendors to extend the standardized management model with vendor-specific extensions to support additional features offered by their products.

#### 5.1.2. Extension Identification

It MUST be possible for a management station to understand which parts of returned data are specific to a given vendor or other standardized extension. The data returned needs to be appropriately marked, through the use of name spaces or similar mechanisms, to ensure that the base management model data can be logically separated from the extension data without needing to understand the extension data itself.

#### 5.1.3. Name-Space Collision Protection

It MUST be possible to protect against multiple extensions conflicting with each other. The use of name-space protection mechanisms for communicated management variables is common practice to protect against such problems. Name-space identification techniques also frequently solve the "Extension Identification" requirement discussed in Section 5.1.2.

## 6. Security Considerations

Any management protocol for which conformance to this document is claimed needs to fully support the criteria discussed in Section 4 in order to protect the management infrastructure itself. The DNS is a core Internet service, and management traffic that protects it could be the target of attacks designed to subvert that service. Because the management infrastructure will be adding additional interfaces to that service, it is critical that the management infrastructure support adequate protections against network attacks.

## 7. Document History

A requirement-gathering discussion was held at the December 2007 IETF meeting in Vancouver, BC, Canada, and a follow-up meeting was held at the March 2008 IETF meeting in Philadelphia. This document is a compilation of the results of those discussions as well as discussions on the DCOMA mailing list.

## 8. Acknowledgments

This document is the result of discussions within the DCOMA design team chaired by Jaap Akkerhuis. This team consisted of a large number of people, all of whom provided valuable insight and input into the discussions surrounding name server management. The text of this document was written from notes taken during meetings as well as from contributions sent to the DCOMA mailing list. This work documents the consensus of the DCOMA design team.

In particular, the following team members contributed significantly to the text in the document:

Stephane Bortzmeyer
Stephen Morris
Phil Regnauld

Further editing contributions and wording suggestions were made by Alfred Hoenes and Doug Barton.

## 9. References

### 9.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

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

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, May 2006.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", RFC 5011, September 2007.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.

## 9.2. Informative References

[RFC1611] Austein, R. and J. Saperia, "DNS Server MIB Extensions", RFC 1611, May 1994.

[RFC1612] Austein, R. and J. Saperia, "DNS Resolver MIB Extensions", RFC 1612, May 1994.

[RFC2182] Elz, R., Bush, R., Bradner, S., and M. Patton, "Selection and Operation of Secondary DNS Servers", BCP 16, RFC 2182, July 1997.

[RFC3197] Austein, R., "Applicability Statement for DNS MIB Extensions", RFC 3197, November 2001.

## Appendix A. Deployment Scenarios

This appendix documents some additional deployment scenarios that have been traditionally difficult to manage. They are provided as guidance to protocol developers as data points of real-world name server management problems.

### A.1. Non-Standard Zones

If an organization uses non-standard zones (for example a purely local TLD), synchronizing all the name servers for these zones is usually a time-consuming task. It is made worse when two organizations with conflicting zones merge. This situation is not a recommended deployment scenario (and is even heavily discouraged), but it is, unfortunately, seen in the wild.

It is typically implemented using "forwarding" zones. But there is no way to ensure automatically that all the resolvers have the same set of zones to forward at any given time. New zones might be added to a local forwarding recursive server, for example, without modifying the rest of the deployed forwarding servers. It is hoped that a management solution that could handle the configuration of zone forwarding would finally allow management of servers deployed in this fashion.

### A.2. Redundancy Sharing

For reliability reasons, it is recommended that zone operators follow the guidelines documented in [RFC2182], which recommends that multiple name servers be configured for each zone and that the name servers be separated both physically and via connectivity routes. A common solution is to establish DNS-serving partnerships: "I'll host your zones and you'll host mine". Both entities benefit from increased DNS reliability via the wider service distribution. This frequently occurs between cooperating but otherwise unrelated entities (such as between two distinct companies) as well as between affiliated organizations (such as between branch offices within a single company).

The configuration of these relationships are currently required to be manually configured and maintained. Changes to the list of zones that are cross-hosted are manually negotiated between the cooperating network administrators and configured by hand. A management protocol with the ability to provide selective authorization, as discussed in Section 4.4, would solve many of the management difficulties between cooperating organizations.

### A.3. DNSSEC Management

There are many different DNSSEC deployment strategies that may be used for mission-critical zones. The following list describes some example deployment scenarios that might warrant different management strategies.

All contents and DNSSEC keying information controlled and operated by a single organization

Zone contents controlled and operated by one organization, all DNSSEC keying information controlled and operated by a second organization.

Zone contents controlled and operated by one organization, zone signing keys (ZSKs) controlled and operated by a second organization, and key signing keys (KSKs) controlled and operated by a third organization.

Although this list is not exhaustive in the potential ways that zone data can be divided up, it should be sufficient to illustrate the potential ways in which zone data can be controlled by multiple entities.

The end result of all of these strategies, however, will be the same: a live zone containing DNSSEC-related resource records. Many of the above strategies are merely different ways of preparing a zone for serving. A management solution that includes support for managing DNSSEC zone data may wish to take into account these potential management scenarios.

## Author's Address

Wes Hardaker
Sparta, Inc.
P.O. Box 382
Davis, CA 95617
US

Phone: +1 530 792 1913
EMail: ietf@hardakers.net