Network Working Group
Request for Comments: 4892
Category: Informational

S. Woolf
Internet Systems Consortium, Inc.
D. Conrad
ICANN

June 2007

Requirements for a Mechanism Identifying a Name Server Instance

Status of This Memo

Copyright Notice

Abstract

With the increased use of DNS anycast, load balancing, and other mechanisms allowing more than one DNS name server to share a single IP address, it is sometimes difficult to tell which of a pool of name servers has answered a particular query. A standardized mechanism to determine the identity of a name server responding to a particular query would be useful, particularly as a diagnostic aid for administrators. Existing ad hoc mechanisms for addressing this need have some shortcomings, not the least of which is the lack of prior analysis of exactly how such a mechanism should be designed and deployed. This document describes the existing convention used in some widely deployed implementations of the DNS protocol, including advantages and disadvantages, and discusses some attributes of an improved mechanism.

1. Introduction and Rationale

Identifying which name server is responding to queries is often useful, particularly in attempting to diagnose name server difficulties. This is most obviously useful for authoritative nameservers in the attempt to diagnose the source or prevalence of inaccurate data, but can also conceivably be useful for caching resolvers in similar and other situations. Furthermore, the ability to identify which server is responding to a query has become more useful as DNS has become more critical to more Internet users, and as network and server deployment topologies have become more complex.

The conventional means for determining which of several possible servers is answering a query has traditionally been based on the use of the server's IP address as a unique identifier. However, the modern Internet has seen the deployment of various load balancing, fault-tolerance, or attack-resistance schemes such as shared use of unicast IP addresses as documented in [RFC3258]. An unfortunate side effect of these schemes has been to make the use of IP addresses as identifiers associated with DNS (or any other) service somewhat problematic. Specifically, multiple dedicated DNS queries may not go to the same server even though sent to the same IP address. Non-DNS methods such as ICMP ping, TCP connections, or non-DNS UDP packets (such as those generated by tools like "traceroute"), etc., may well be even less certain to reach the same server as the one which receives the DNS queries.

There is a well-known and frequently-used technique for determining an identity for a nameserver more specific than the possibly-non-unique "server that answered the query I sent to IP address A.B.C.D". The widespread use of the existing convention suggests a need for a documented, interoperable means of querying the identity of a nameserver that may be part of an anycast or load-balancing cluster. At the same time, however, it also has some drawbacks that argue against standardizing it as it's been practiced so far.

2. Existing Conventions

For some time, the commonly deployed Berkeley Internet Name Domain (BIND) implementation of the DNS protocol suite from the Internet Systems Consortium [BIND] has supported a way of identifying a particular server via the use of a standards-compliant, if somewhat unusual, DNS query. Specifically, a query to a recent BIND server for a TXT resource record in class 3 (CHAOS) for the domain name "HOSTNAME.BIND." will return a string that can be configured by the name server administrator to provide a unique identifier for the responding server. (The value defaults to the result of a gethostname() call). This mechanism, which is an extension of the BIND convention of using CHAOS class TXT RR queries to sub-domains of the "BIND." domain for version information, has been copied by several name server vendors.

A refinement to the BIND-based mechanism, which dropped the implementation-specific label, replaces "BIND." with "SERVER.". Thus the query label to learn the unique name of a server may appear as "ID.SERVER.".

(For reference, the other well-known name used by recent versions of BIND within the CHAOS class "BIND." domain is "VERSION.BIND.". A query for a CHAOS TXT RR for this name will return an

administratively defined string which defaults to the software version of the server responding. This is, however, not generally implemented by other vendors.)

2.1. Advantages

There are several valuable attributes to this mechanism, which account for its usefulness.

1. The "HOSTNAME.BIND." or "ID.SERVER." query response mechanism is within the DNS protocol itself. An identification mechanism that relies on the DNS protocol is more likely to be successful (although not guaranteed) in going to the same system as a "normal" DNS query.

2. Since the identity information is requested and returned within the DNS protocol, it doesn't require allowing any other query mechanism to the server, such as holes in firewalls for otherwise-unallowed ICMP Echo requests. Thus it is likely to reach the same server over a path subject to the same routing, resource, and security policy as the query, without any special exceptions to site security policy.

3. It is simple to configure. An administrator can easily turn on this feature and control the results of the relevant query.

4. It allows the administrator complete control of what information is given out in the response, minimizing passive leakage of implementation or configuration details. Such details are often considered sensitive by infrastructure operators.

2.2. Disadvantages

At the same time, there are some serious drawbacks to the CHAOS/TXT query mechanism that argue against standardizing it as it currently operates.

1. It requires an additional query to correlate between the answer to a DNS query under normal conditions and the supposed identity of the server receiving the query. There are a number of situations in which this simply isn't reliable.

2. It reserves an entire class in the DNS (CHAOS) for what amounts to one zone. While CHAOS class is defined in [RFC1034] and [RFC1035], it's not clear that supporting it solely for this purpose is a good use of the namespace or of implementation effort.

3. The initial and still common form, using "BIND.", is implementation specific. BIND is one DNS implementation. At the time of this writing, it is probably most prevalent for authoritative servers. This does not justify standardizing on its ad hoc solution to a problem shared across many operators and implementors. Meanwhile, the aforementioned refinement changes the query label but preserves the ad hoc CHAOS/TXT mechanism.

4. There is no convention or shared understanding of what information an answer to such a query for a server identity could or should contain, including a possible encoding or authentication mechanism.

5. Hypothetically, since DNSSEC has been defined to cover all DNS classes, the TXT RRs returned in response to the "ID.SERVER." query could be signed, which has the advantages described in [RFC4033]. However, since DNSSEC deployment for the CHAOS class is neither existent nor foreseeable, and since the "ID.SERVER." TXT RR is expected to be unique per server, this would be impossible in practice.

The first of the listed disadvantages may be technically the most serious. It argues for an attempt to design a good answer to the problem, "I need to know what nameserver is answering my queries", not simply a convenient one.

3. Characteristics of an Implementation Neutral Convention

The discussion above of advantages and disadvantages to the "HOSTNAME.BIND." mechanism suggest some requirements for a better solution to the server identification problem. These are summarized here as guidelines for any effort to provide appropriate protocol extensions:

1. The mechanism adopted must be in-band for the DNS protocol. That is, it needs to allow the query for the server’s identifying information to be part of a normal, operational query. It should also permit a separate, dedicated query for the server’s identifying information. But it should preserve the ability of the CHAOS/TXT query-based mechanism to work through firewalls and in other situations where only DNS can be relied upon to reach the server of interest.

2. The new mechanism should not require dedicated namespaces or other reserved values outside of the existing protocol mechanisms for these, i.e., the OPT pseudo-RR. In particular, it should not propagate the existing drawback of requiring support for a CLASS

and top level domain in the authoritative server (or the querying tool) to be useful.

3. Support for the identification functionality should be easy to implement and easy to enable. It must be easy to disable and should lend itself to access controls on who can query for it.

4. It should be possible to return a unique identifier for a server without requiring the exposure of information that may be non-public and considered sensitive by the operator, such as a hostname or unicast IP address maintained for administrative purposes.

5. It should be possible to authenticate the received data by some mechanism analogous to those provided by DNSSEC. In this context, the need could be met by including encryption options in the specification of a new mechanism.

6. The identification mechanism should not be implementation-specific.

4. IANA Considerations

This document proposes no specific IANA action. Protocol extensions, if any, to meet the requirements described are out of scope for this document. A proposed extension, specified and adopted by normal IETF process, is described in [NSID], including relevant IANA action.

5. Security Considerations

Providing identifying information as to which server is responding to a particular query from a particular location in the Internet can be seen as information leakage and thus a security risk. This motivates the suggestion above that a new mechanism for server identification allow the administrator to disable the functionality altogether or partially restrict availability of the data. It also suggests that the server identification data should not be readily correlated with a hostname or unicast IP address that may be considered private to the nameserver operator’s management infrastructure.

Propagation of protocol or service meta-data can sometimes expose the application to denial of service or other attack. As the DNS is a critically important infrastructure service for the production Internet, extra care needs to be taken against this risk for designers, implementors, and operators of a new mechanism for server identification.

Both authentication and confidentiality of server identification data are potentially of interest to administrators -- that is, operators may wish to make such data available and reliable to themselves and their chosen associates only. This constraint would imply both an ability to authenticate it to themselves and to keep it private from arbitrary other parties, which leads to characteristics 4 and 5 of an improved solution.

6. Acknowledgements

The technique for host identification documented here was initially implemented by Paul Vixie of the Internet Software Consortium in the Berkeley Internet Name Daemon package. Comments and questions on earlier versions were provided by Bob Halley, Brian Wellington, Andreas Gustafsson, Ted Hardie, Chris Yarnell, Randy Bush, and members of the ICANN Root Server System Advisory Committee. The newest version takes a significantly different direction from previous versions, owing to discussion among contributors to the DNSOP working group and others, particularly Olafur Gudmundsson, Ed Lewis, Bill Manning, Sam Weiler, and Rob Austein.

7. References

7.1. 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.

[RFC3258] Hardie, T., "Distributing Authoritative Name Servers via Shared Unicast Addresses", RFC 3258, April 2002.

7.2. Informative References

[BIND] ISC, "BIND 9 Configuration Reference".

[NSID] Austein, R., "DNS Name Server Identifier Option (NSID)", Work in Progress, June 2006.

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

Authors' Addresses

Suzanne Woolf
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
US

Phone: +1 650 423-1333
EMail: woolf@isc.org
URI: http://www.isc.org/

David Conrad
ICANN
4676 Admiralty Way
Marina del Rey, CA 90292
US

Phone: +1 310 823 9358
EMail: david.conrad@icann.org
URI: http://www.iana.org/

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.