Internet Engineering Task Force
Internet-Draft

P. Barber
J. Brady
M. Larson
VeriSign, Inc.

November 9, 2001

Expires: May 9, 2002

Observed DNS Resolution Misbehavior

<draft-ietf-dnsop-bad-dns-res-00.txt>

Status of this memo

Abstract

This Internet-Draft describes DNS name server and stub resolver behavior that results in a significant query volume sent to the root and top-level domain (TLD) name servers. In some cases we recommend minor additions to the DNS protocol specification and corresponding changes in name server implementations to alleviate these unnecessary queries. In one case, we have highlighted behavior of a popular name server implementation that does not conform to the DNS specification. The recommendations made in this document are a direct byproduct of observation and analysis of abnormal query traffic patterns seen at two of the thirteen root name servers and all thirteen com/net/org TLD name servers.

Conventions used in this document

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].

Table of contents

1. Introduction

Observation of query traffic received by two root name servers and the thirteen com/net/org TLD name servers has revealed that a large proportion of the total traffic often consists of "requeries". A requery is the same question (<qname, qtype, qclass>) asked repeatedly at an unexpectedly high rate. We have observed requeries from both a single IP address and multiple IP addresses.

By analyzing requery events we have found that the cause of the duplicate traffic is almost always a deficient name server, stub resolver and/or application implementation combined with an operational anomaly. The implementation deficiencies we have identified to date include well-intentioned recovery attempts gone awry, insufficient caching of failures, early abort when multiple levels of glue records must be followed, and aggressive retry by stub resolvers and/or applications. Anomalies that we have seen trigger requery events include lame delegations, unusual glue records, and anything that makes all authoritative name servers for a zone unreachable (DoS attacks, crashes, maintenance, routing failures, congestion, etc.).

In the following sections, we provide a detailed explanation of the observed behavior and recommend changes that will reduce the requery rate. Some of the changes recommended affect the core DNS protocol specification, described principally in [RFC1034], [RFC1035] and [RFC2181].

2. Observed name server misbehavior

2.1 Aggressive requerying for delegation information

There can be times when every name server in a zone's NS RRset is unreachable (e.g., during a network outage), unavailable (e.g., the name server process is not running on the server host) or misconfigured (e.g., the name server is not authoritative for the given zone, also known as "lame"). Consider a name server that attempts to resolve a recursive query for a domain name in such a zone and discovers that none of the zone's name servers can provide an answer. We have observed a recursive name server implementation that then verifies the zone's NS RRset in its cache by querying for the zone's delegation information: it sends a query for the zone's NS RRset to one of the parent zone's name servers.

For example, suppose that example.com has the following NS RRset:

```
example.com.   IN   NS   ns1.example.com.
example.com.   IN   NS   ns2.example.com.
```

Upon receipt of a query for www.example.com and assuming that neither ns1.example.com nor ns2.example.com can provide an answer, this recursing name server implementation immediately queries a com zone name server for the example.com NS RRset to verify it has the proper delegation information. This name server implementation performs this query to a zone's parent zone for each recursive query it receives that fails because of a completely unresponsive set of name servers for the target zone. Consider the effect when a popular zone experiences a catastrophic failure of all its name servers: now every recursive query for domain names in that zone sent to this name server implementation results in a query to the failed zone's parent name servers. On one occasion when several dozen popular zones became unreachable, the query load to the com/net/org name servers increased by 50%.

We believe this verification query is not reasonable. Consider the circumstances: When a recursing name server is resolving a query for a domain name in a zone it has not previously searched, it uses the list of name servers in the referral from the target zone's parent. If on its first attempt to search the target zone, none of the name servers in the referral are reachable, a verification query to the parent is pointless: this query to the parent would come so quickly on the heels of the referral that it would be almost certain to contain the same list of name servers. The chance of discovering any new information is slim.

The other possibility is that the recursing name server

successfully contacts one of the target zone's name servers and then caches the NS RRset from the authority section of a response, the proper behavior according to section 5.4.1 of [RFC2181], because the NS RRset from the target zone is more trustworthy than delegation information from the parent zone. If, while processing a subsequent recursive query, the recursing name server discovers that none of the name servers specified in the cached NS RRset is available or authoritative, querying the parent would be wrong. An NS RRset from the parent zone would now be less trustworthy than data already in the cache.

For this query of the parent zone to be useful, the target zone's entire set of name servers would have to change AND the former set of name servers would have to be deconfigured and/or decomissioned AND the delegation information in the parent zone would have to be updated with the new set of name servers, all within the TTL of the target zone's NS RRset. We believe this scenario is uncommon: administrative best practices dictate that changes to a zone's set of name servers happen gradually, with servers that are removed from the NS RRset left authoritative for the zone as long as possible. The scenarios that we can envision that would benefit from the parent requery behavior do not outweigh its damaging effects.

2.1.1 Recommendation

Name servers offering recursion MUST NOT send a query for the NS RRset of a non-responsive zone to any of the name servers for that zone's parent zone. For the purposes of this injunction, a non-responsive zone is defined as a zone for which every name server listed in the zone's NS RRset:

(1) is not authoritative for the zone (i.e., lame), or,
(2) returns a server failure response (SERVFAIL), or,
(3) is dead or unreachable according to section 7.2 of [RFC2308].

2.2 Repeated queries to lame servers

Section 2.1 describes a catastrophic failure: when every name server for a zone is unable to provide an answer for one reason or another. A more common occurrence is a subset of a zone's name servers being unavailable or misconfigured. Different failure modes have different expected durations. Some symptoms indicate problems that are potentially transient: various types of ICMP unreachable messages because a name server process is not running or a host or network is unreachable, or a complete lack of a response to a query. Such responses could be the result of a host rebooting or temporary outages; these events don't necessarily

require any human intervention and can be reasonably expected to be temporary.

Other symptoms clearly indicate a condition requiring human intervention, such as lame server: if a name server is misconfigured and not authoritative for a zone delegated to it, it is reasonable to assume that this condition has potential to last longer than unreachability or unresponsiveness. Consequently, repeated queries to known lame servers are not useful. In this case of a condition with potential to persist for a long time, a better practice would be to maintain a list of known lame servers and avoid querying them repeatedly in a short interval.

2.2.1 Recommendation

Name servers offering recursion SHOULD cache name servers that they discover are not authoritative for zones delegated to them (i.e. lame servers). Lame servers MUST be cached against the specific query tuple <zone name, class, server IP address>. Zone name can be derived from the owner name of the NS record that was referenced to query the name server that was discovered to be lame. Implementations that perform lame server caching MUST refrain from sending queries to known lame servers based on a time interval from when the server is discovered to be lame. A minimum interval of thirty minutes is RECOMMENDED.

2.3 Incomplete negative caching implementation

A widely distributed name server implementation does not properly implement negative caching as described in [RFC2308]. In particular, this implementation does not cache NODATA responses. Such a response indicates that the queried domain name exists but has no records of the desired type. See section 2.2 of [RFC2308] for information on how NODATA responses are indicated.

2.3.1 Recommendation

Vendors of any name server implementations that do not comply with [RFC2308] are encouraged to bring their software into conformance.

2.4 Inability to follow multiple levels of out-of-zone glue

Some name server implementations are unable to follow more than one level of out-of-zone glue. For example, consider the following delegations:

```
foo.example.        IN   NS   ns1.example.com.
foo.example.        IN   NS   ns2.example.com.
```

```
example.com.        IN   NS   ns1.test.example.net.
example.com.        IN   NS   ns2.test.example.net.

test.example.net.   IN   NS   ns1.test.example.net.
test.example.net.   IN   NS   ns2.test.example.net.
```

A name server processing a recursive query for www.foo.example must follow two levels of indirection, first obtaining address records for ns1.test.example.net and/or ns2.test.example.net in order to obtain address records for ns1.example.com and/or ns2.example.com in order to query those name servers for the address records of www.foo.example. While this situation may appear contrived, we have seen multiple similar occurrences and expect more as the new generic top-level domains (gTLDs) become active. We anticipate many zones in the new gTLDs will use name servers in other gTLDs, increasing the amount of inter-zone glue.

2.4.1 Recommendation

Certainly constructing a delegation that relies on multiple levels of out-of-zone glue is not a good administrative practice. This issue could be mitigated with an operational injunction in an RFC to refrain from construction of such delegations. In our opinion the practice is widespread enough to merit clarifications to the DNS protocol specification to permit it on a limited basis.

Name servers offering recursion SHOULD be able to handle at least three levels of indirection resulting from out-of-zone glue.

3. Observed client misbehavior

We have observed situations where a zone's name servers are misconfigured or unavailable, resulting in a SERVFAIL response from a recursive name server in response to queries for domain names in that zone. In some instances, we then observe many repeated queries (on the order of hundreds per second) to the com/net/org name servers for domain names in the affected zones. Sometimes the queries originate from multiple source IP addresses, while at other times a single source address sends many repeated queries. This behavior appears to be triggered by a SERVFAIL response (i.e., upon investigation, the <qname, qtype, qclass> of a repeated query at the com/net/org name servers produces a SERVFAIL response when sent to a local recursive name server.)

We suspect that some DNS clients (i.e., stub resolvers) and/or application programs have overzealous retransmission algorithms that are trigged by a SERVFAIL response. Unfortunately, we have not been able to isolate particular implementations. The authors encourage and welcome reports of DNS clients and applications with overzealous retransmission algorithms.

4. IANA considerations

There are no new IANA considerations introduced by this Internet-Draft.

5. Security considerations

Nameserver, stub resolver and application misbehaviors identical or similar to those observed and discussed in this document expose root and TLD name server constellations to increased risk of both intentional and unintentional denial of service.

We believe that implementation of the recommendations offered in this document will reduce the requery traffic seen at root and TLD name servers, thus reducing the opportunity for an attacker to use such requerying to his or her advantage.

6. Internationalization considerations

We do not believe this document introduces any new internationalization considerations to the DNS protocol specification.

7. References

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

[RFC1035] - Domain Names - Implementation and Specifications, P. Mockapetris, November 1987.

[RFC2119] - Key Words for Use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997.

[RFC2181] - Clarifications to the DNS Specification, R. Elz, R. Bush, July 1997.

[RFC2308] - Negative Caching of DNS Queries (DNS NCACHE), M. Andrews, March 1998.

8. Authors' addresses

Piet Barber
VeriSign Global Registry Services
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA
pbarber@verisign.com

John Brady
VeriSign Global Registry Services
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA
jbrady@verisign.com

Matt Larson
VeriSign Global Registry Services
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA
mlarson@verisign.com

A. Full copyright statement

Acknowledgement

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