Internet Engineering Task Force (IETF)
Internet-Draft
Intended status: Informational
Expires: October 21, 2018

L. Song, Ed.
D. Liu
Beijing Internet Institute
P. Vixie
TISF
Kato
Keio University/WIDE Project
S. Kerr
April 19, 2018

Yeti DNS Testbed
draft-song-yeti-testbed-experience-07

Abstract

The Internet's Domain Name System is designed and built on a single root, known as the Root Server System.

Yeti DNS is an experimental, non-production root server testbed that provides an environment where technical and operational experiments can safely be performed without risk to production root server infrastructure.This testbed has been used by a broad community of participants to perform experiments that aim to inform operations and future development of the production DNS. Yeti DNS is an independently-coordinated project and is not affiliated with ICANN, IANA or any Root Server Operator.

The Yeti DNS testbed implementation includes various novel and experimental components including IPv6-only transport, independent, autonomous Zone Signing Key management, large cryptographic keys and a large number of Yeti-Root Servers names. These differences from the Root Server System have operational consequences such as large responses to priming queries and the coordination of a large pool of independent operators; by deploying such a system globally but outside the production DNS system, the Yeti DNS project provides an opportunity to gain insight into those consequences without threatening the stability of the DNS.

This document neither addresses the relevant policies under which the Root Server System is operated nor makes any proposal for changing any aspect of its implementation or operation. This document aims solely to document the technical and operational experience of deploying a system which is similar to but different from the Root Server System.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Domain Name System (DNS), as originally specified in [RFC1034] and [RFC1035], has proved to be an enduring and important platform upon which almost every end-user of the Internet relies. Despite its longevity, extensions to the protocol, new implementations and refinements to DNS operations continue to emerge both inside and outside the IETF.

The Root Server System in particular has seen technical innovation and development, for example in the form of wide-scale anycast deployment, the mitigation of unwanted traffic on a global scale, the widespread deployment of Response Rate Limiting [RRL], the introduction of IPv6 transport, the deployment of DNSSEC, changes in DNSSEC key sizes, and preparations to roll the root zone's Key Signing Key (KSK) and corresponding trust anchor. These projects created tremendous qualitative operational change, and required impressive caution and study prior to implementation. They took place in parallel with the quantitative expansion or delegations for new TLDs. <https://newgtlds.icann.org/>.

Aspects of the operational structure of the Root Server System have been described in such documents as [TNO2009], [ISC-TN-2003-1], [RSSAC001] and [RFC7720]. Such references, considered together, provide sufficient insight into the operations of the system as a whole that it is straightforward to imagine structural changes to the root server system's infrastructure and to wonder what the operational implications of such changes might be.

The Yeti DNS Project was conceived in May 2015 to provide a non-production testbed upon which the technical community could propose and run experiments designed to answer these kinds of questions. Coordination for the project was provided by TISF, the WIDE Project and the Beijing Internet Institute. Many volunteers collaborated to build a distributed testbed that at the time of writing includes 25 Yeti root servers with 16 operators and handles experimental traffic from individual volunteers, universities, DNS vendors and distributed measurement networks.

By design, the Yeti testbed system serves the root zone published by the IANA with only those structural modifications necessary to ensure that it is able to function usefully in the Yeti testbed system instead of the production Root Server system. In particular, no delegation for any top-level zone is changed, added or removed from the IANA-published root zone to construct the root zone served by the Yeti testbed system, and changes in the root zone are reflected in the testbed in near real-time. In this document, for clarity, we refer to the zone derived from the IANA-published root zone as the Yeti-Root zone.

The Yeti DNS testbed serves a similar function to the Root Server System in the sense that they both serve similar zones: the Yeti-Root zone and the IANA-published root zone. However, the Yeti DNS testbed only serves clients that are explicitly configured to participate in the experiment, whereas the Root Server System serves the whole Internet. Since the dependent end-users and systems of the Yeti DNS testbed are known and their operations well-coordinated with those of

the Yeti project, it has been possible to deploy structural changes in the Yeti DNS testbed with effective measurement and analysis, something that is difficult or simply impractical in the production Root Server System.

2. Requirements Notation and Conventions

Through the document, any mention to "Root" with an uppercase R and without other prefix, refers to the "IANA Root" systems used in the production Internet. Proper mentions to the Yeti infrastructure will be prefixed with "Yeti", like "Yeti-Root Zone", "Yeti-DNS", and so on.

3. Areas of Study

Examples of topics that the Yeti DNS Testbed was built to address are included below, each illustrated with indicative questions.

3.1. Implementation of a Root Server System-like Testbed

- o How can a testbed be constructed and deployed on the Internet, allowing useful public participation without any risk of disruption of the Root Server System?

- o How can representative traffic be introduced into such a testbed such that insights into the impact of specific differences between the testbed and the Root Server System can be observed?

3.2. Yeti-Root Zone Distribution

- o What are the scaling properties of Yeti-Root zone distribution as the number of Yeti-Root servers, Yeti-Root server instances or intermediate distribution points increase?

3.3. Yeti-Root Server Names and Addressing

- o What naming schemes other than those closely analogous to the use of ROOT-SERVERS.NET in the production root zone are practical, and what are their respective advantages and disadvantages?

- o What are the risks and benefits of signing the zone that contains the names of the Yeti-Root servers?

- o What automatic mechanisms might be useful to improve the rate at which clients of Yeti-Root servers are able to react to a Yeti-Root server renumbering event?

3.4. IPv6-Only Yeti-Root Servers

- o Are there negative operational effects in the use of IPv6-only Yeti-Root servers, compared to the use of servers that are dual-stack?

- o What effect does the IPv6 fragmentation model have on the operation of Yeti-Root servers, compared with that of IPv4?

3.5. DNSSEC in the Yeti-Root Zone

- o Is it practical to sign the Yeti-Root zone using multiple, independently-operated DNSSEC signers and multiple corresponding ZSKs?

- o To what extent is [RFC5011]: "Automated Updates of DNS Security (DNSSEC) Trust Anchors" supported by resolvers?

- o Does the KSK Rollover plan designed and in the process of being implemented by ICANN work as expected on the Yeti testbed?

- o What is the operational impact of using much larger RSA key sizes in the ZSKs used in a root?

- o What are the operational consequences of choosing DNSSEC algorithms other than RSA to sign a root?

4. Yeti DNS Testbed Infrastructure

The purpose of the testbed is to allow DNS queries from stub resolvers, mediated by recursive resolvers, to be delivered to Yeti-Root servers, and for corresponding responses generated on the Yeti-Root servers to be returned, as illustrated in Figure 1.

```
 ,----------.         ,-----------.         ,------------.
 |   stub   +------> | recursive +------> | Yeti-Root  |
 | resolver | <------+ resolver  | <------+ nameserver |
 `----------'         `-----------'         `------------'
    ^                    ^                     ^
    |  appropriate       |  Yeti-Root hints;   |  Yeti-Root zone
    `- resolver          `- Yeti-Root trust    `- with DNSKEY RRSet
       configured           anchor                signed by
                                                  Yeti-Root KSK
```

Figure 1: High-Level Testbed Components

To use the Yeti DNS testbed, a recursive resolver must be configured to use the Yeti-Root servers. That configuration consists of a list of names and addresses for the Yeti-Root servers (often referred to as a "hints file") that replaces the corresponding hints used for the production Root Server System (Appendix A). If resolvers are configured to validate DNSSEC, then they also need to be configured with a DNSSEC trust anchor that corresponds to a KSK used in the Yeti DNS Project, in place of the normal trust anchor set used for the Root Zone.

Since the Yeti root(s) are signed with Yeti keys, rather than those used by the IANA root, corresponding changes are needed in the resolver trust anchors. Corresponding changes are required in the Yeti-Root hints file Appendix A. Those changes would be properly rejected by any validator using the production Root Server System's root zone trust anchor set as bogus.

Stub resolvers become part of the Yeti DNS Testbed by their use of recursive resolvers that are configured as described above.

The data flow from IANA to stub resolvers through the Yeti testbed is illustrated in Figure 2 and are described in more detail in the sections that follow.

```
                                  ,----------------.
                             ,-- / IANA Root Zone / ---.
                             |  `----------------'     |
                             |            |            |
                             |            |            |       Root Zone
     ,--------------.    ,---V---.    ,---V---.    ,---V---.
     | Yeti Traffic |    | BII   |    | WIDE  |    | TISF  |
     | Collection   |    |  DM   |    |  DM   |    |  DM   |
     `----+----+----'    `---+---'    `---+---'    `---+---'
          |    |       ,-----'    ,-------'            `----.
          |    |       |          |                         |  Yeti-Root
          ^    ^       |          |                         |     Zone
          |    |   ,---V---.  ,---V---.                 ,---V---.
          |    `---+ Yeti  |  | Yeti  |  . . . . . . .  | Yeti  |
          |        | Root  |  | Root  |                 | Root  |
          |        `---+---'  `---+---'                 `---+---'
          |            |          |                         |    DNS
          |            |          |                         |  Response
          |         ,--V----------V-------------------------V--.
          `---------+              Yeti Resolvers              |
                    `--------------------+---------------------'
                                         |                       DNS
                                         |                     Response
                    ,--------------------V---------------------.
                    |            Yeti Stub Resolvers           |
                    `------------------------------------------'
```

Figure 2: Testbed Data Flow

Note that the roots are not bound to DMs. DMs update their zone in a time schedule describe in section 3.1. Each of DMs who update the latest zone can send notify to all roots. So the zone transfer can happened between any DM and any root.

4.1. Root Zone Retrieval

The Yeti-Root Zone is distributed within the Yeti DNS testbed through a set of internal master servers that are referred to as Distribution Masters (DMs). These server elements distribute the Yeti-Root zone to all Yeti-Root servers. The means by which the Yeti DMs construct the Yeti-Root zone for distribution is described below.

Since Yeti DNS DMs do not receive DNS NOTIFY [RFC1996] messages from the Root Server System, a polling approach is used to determine when new revisions of the root zone are available from the production Root Server System. Each Yeti DM requests the Root Zone SOA record from a Root server that permits unauthenticated zone transfers of the root

zone, and performs a zone transfer from that server if the retrieved value of SOA.SERIAL is greater than that of the last retrieved zone.

At the time of writing, unauthenticated zone transfers of the Root Zone are available directly from B-Root, C-Root, F-Root, G-Root and K-Root, and from L-Root via the two servers XFR.CJR.DNS.ICANN.ORG and XFR.LAX.DNS.ICANN.ORG, as well as via FTP from sites maintained by the Root Zone Maintainer and the IANA Functions Operator. The Yeti DNS Testbed retrieves the Root Zone using zone transfers from F-Root. The schedule on which F-Root is polled by each Yeti DM is as follows:

DM Operator	Time
BII	UTC hour + 00 minutes
WIDE	UTC hour + 20 minutes
TISF	UTC hour + 40 minutes

The Yeti DNS testbed uses multiple DMs, each of which acts autonomously and equivalently to its siblings. Any single DM can act to distribute new revisions of the Yeti-Root zone, and is also responsible for signing the RRSets that are changed as part of the transformation of the Root Zone into the Yeti-Root zone described in Section 4.2. This multiple DM model intend to provide a basic structure to implement idea of shared zone control proposed in [ITI2014].

4.2. Transformation of Root Zone to Yeti-Root Zone

Two distinct approaches have been deployed in the Yeti-DNS Testbed, separately, to transform the Root Zone into the Yeti-Root Zone. At a high level both approaches are equivalent in the sense that they replace a minimal set of information in the root zone with corresponding data for the Yeti DNS Testbed; the mechanisms by which the transforms are executed are different, however. Each is discussed in turn in Section 4.2.1 and Section 4.2.2, respectively.

A third approach has also been proposed, but not yet implemented. The motivations and changes implied by that approach are described in Section 5.2.3.

4.2.1. ZSK and KSK Key Sets Shared Between DMs

The approach described here was the first to be implemented. It features entirely autonomous operation of each DM, but also requires secret key material (the private key in each of the Yeti-Root KSK and

ZSK key-pairs) to be distributed and maintained on each DM in a coordinated way.

The Root Zone is transformed as follows to produce the Yeti-Root Zone. This transformation is carried out autonomously on each Yeti DNS Project DM. Each DM carries an authentic copy of the current set of Yeti KSK and ZSK key pairs, synchronized between all DMs (see Section 4.4).

1. SOA.MNAME is set to www.yeti-dns.org.

2. SOA.RNAME is set to <dm-operator>.yeti-dns.org. where <dm-operator> is currently one of "wide", "bii" or "tisf".

3. All DNSKEY, RRSIG and NSEC records are removed.

4. The apex NS RRSet is removed, with the corresponding root server glue (A and AAAA) RRSets.

5. A Yeti DNSKEY RRSet is added to the apex, comprising the public parts of all Yeti KSK and ZSKs.

6. A Yeti NS RRSet is added to the apex that includes all Yeti-Root servers.

7. Glue records (AAAA only, since Yeti-Root servers are v6-only) for all Yeti-Root servers are added.

8. The Yeti-Root Zone is signed: the NSEC chain is regenerated; the Yeti KSK is used to sign the DNSKEY RRSet, and the shared ZSK is used to sign every other RRSet.

Note that the SOA.SERIAL value published in the Yeti-Root Zone is identical to that found in the root zone.

4.2.2. Unique ZSK per DM; No Shared KSK

The approach described here was the second to be implemented and maintained as stable state. Each DM is provisioned with its own, dedicated ZSK key pairs that are not shared with other DMs. A Yeti-Root DNSKEY RRSet is constructed and signed upstream of all DMs as the union of the set of active Yeti-Root KSKs and the set of active ZSKs for every individual DM. Each DM now only requires the secret part of its own dedicated ZSK key pairs to be available locally, and no other secret key material is shared. The high-level approach is illustrated in Figure 3.

```
                       ,----------.         ,-----------.
              .--------> BII ZSK  +---------> Yeti-Root |
              | signs  `----------'  signs  `-----------'
              |
 ,-----------. |        ,----------.         ,-----------.
 | Yeti KSK  +-+--------> TISF ZSK +---------> Yeti-Root |
 `-----------' | signs  `----------'  signs  `-----------'
              |
              |        ,----------.         ,-----------.
              `--------> WIDE ZSK +---------> Yeti-Root |
                signs  `----------'  signs  `-----------'
```

Figure 3: Unique ZSK per DM

The process of retrieving the Root Zone from the Root Server System and replacing and signing the apex DNSKEY RRSet no longer takes place on the DMs, and instead takes place on a central Hidden Master. The production of signed DNSKEY RRSets is analogous to the use of Signed Key Responses (SKR) produced during ICANN KSK key ceremonies [ICANN2010].

Each DM now retrieves source data (with pre-modified and Yeti-signed DNSKEY RRset, but otherwise unchanged) from the Yeti DNS Hidden Master instead of from the Root Server System.

Each DM carries out a similar transformation to that described in Section 4.2.1, except that DMs no longer need to modify or sign the DNSKEY RRSet, and the DM's unique local ZSK is used to sign every other RRset.

4.2.3. Preserving Root Zone NSEC Chain and ZSK RRSIGs

A change to the transformation described in Section 4.2.2 has been proposed that would preserve the NSEC chain from the Root Zone and all RRSIG RRs generated using the Root Zone's ZSKs. The DNSKEY RRSet would continue to be modified to replace the Root Zone KSKs, but Root Zone ZSKs will be kept intact, and the Yeti KSK would be used to generate replacement signatures over the apex DNSKEY and NS RRSets. Source data would continue to flow from the Root Server System through the Hidden Master to the set of DMs, but no DNSSEC operations would be required on the DMs and the source NSEC and most RRSIGs would remain intact.

This approach has been suggested in order to provide cryptographically-verifiable confidence that no owner name in the root zone had been changed in the process of producing the Yeti-Root

zone from the Root Zone, addressing one of the concerns described in Appendix E in a way that can be verified automatically.

4.3. Yeti-Root Zone Distribution

Each Yeti DM is configured with a full list of Yeti-Root Server addresses to send NOTIFY [RFC1996] messages to, which also forms the basis for an address-based access-control list for zone transfers. Authentication by address could be replaced with more rigourous mechanisms (e.g. using Transaction Signatures (TSIG) [RFC2845]); this has not been done at the time of writing since the use of address-based controls avoids the need for the distribution of shared secrets amongst the Yeti-Root Server Operators.

Individual Yeti-Root Servers are configured with a full set of Yeti DM addresses to which SOA and AXFR queries may be sent in the conventional manner.

4.4. Synchronization of Service Metadata

Changes in the Yeti-DNS Testbed infrastructure such as the addition or removal of Yeti-Root servers, renumbering Yeti-Root Servers or DNSSEC key rollovers require coordinated changes to take place on all DMs. The Yeti-DNS Testbed is subject to more frequent changes than are observed in the Root Server System and includes substantially more Yeti-Root Servers than there are IANA Root Servers, and hence a manual change process in the Yeti Testbed would be more likely to suffer from human error. An automated and cooperated process was consequently implemented.

The theory of this operation is that each DM operator runs a Git repository locally, containing all service metadata involved in the operation of each DM. When a change is desired and approved among all Yeti coordinators, one DM operator (usually BII) updates the local Git repository. A serial number in the future (in two days) is chosen for when the changes become active. The DM operator then pushes the changes to the Git repositories of the other two DM operators who have chance to check and edit the repo. When the serial of the root zone passes the number chosen, then changes were pulled automatically to individual DMs and promoted to production.

The three Git repositories are synchronized by configuring them as remote servers. For example at BII we push to all three DM's repo as follows:

```
$ git remote -v
origin yeticonf@yeti-conf.dns-lab.net:dm (fetch)
origin yeticonf@yeti-conf.dns-lab.net:dm (push)
origin yeticonf@yeti-dns.tisf.net:dm (push)
origin yeticonf@yeti-repository.wide.ad.jp:dm (push)
```

Figure 4

More detailed information of DM Synchronization, please find the Yeti-DM-Sync-MZSK.md <https://github.com/BII-Lab/Yeti-Project/blob/master/doc/Yeti-DM-Sync-MZSK.md> document in Yeti's GitHub repo.

4.5. Yeti-Root Server Naming Scheme

The current naming scheme for Root Servers was normalized to use single-character host names (A through M) under the domain ROOT-SERVERS.NET, as described in [RSSAC023]. The principal benefit of this naming scheme was that DNS label compression could be used to produce a priming response that would fit within 512 bytes at the time it was introduced, 512 bytes being the maximum DNS message size using UDP transport without EDNS(0) [RFC6891].

Yeti-Root Servers do not use this optimization, but rather use free-form nameserver names chosen by their respective operators -- in other words, no attempt is made to minimize the size of the priming response through the use of label compression. This approach aims to challenge the need for a minimally-sized priming response in a modern DNS ecosystem where EDNS(0) is prevalent.

Priming responses from Yeti-Root Servers do not always include server addresses in the additional section, as is the case with priming responses from Root Servers. In particular, Yeti-Root Servers running BIND9 return an empty additional section if the configuration parameter minimum-responses is set, forcing resolvers to complete the priming process with a set of conventional recursive lookups in order to resolve addresses for each Yeti-Root server. The Yeti-Root Servers running NSD were observed to return a fully-populated additional section (depending of course of the EDNS buffer size in use).

Various approaches to normalize the composition of the priming response were considered, including:

- o Require use of DNS implementations that exhibit a desired behaviour in the priming response;

- o Modify nameserver software or configuration as used by Yeti-Root Servers;

- o Isolate the names of Yeti-Root Servers in one or more zones that could be slaved on each Yeti-Root Server, renaming servers as necessary, giving each a source of authoritative data with which the authority section of a priming response could be fully populated. This is the approach used in the Root Server System with the ROOT-SERVERS.NET zone.

The potential mitigation of renaming all Yeti-Root Servers using a scheme that would allow their names to exist directly in the root zone was not considered, since that approach implies the invention of new top-level labels not present in the Root Zone.

Given the relative infrequency of priming queries by individual resolvers and the additional complexity or other compromises implied by each of those mitigations, the decision was made to make no effort to ensure that the composition of priming responses was identical across servers. Even the empty additional sections generated by Yeti-Root Servers running BIND9 seem to be sufficient for all resolver software tested; resolvers simply perform a new recursive lookup for each authoritative server name they need to resolve.

4.6. Yeti-Root Servers

Various volunteers have donated authoritative servers to act as Yeti-Root servers. At the time of writing there are 25 Yeti-Root servers distributed globally, one of which is named using an IDNA2008 [RFC5890] label, shown in the following list in punycode.

Name	Operator	Location
bii.dns-lab.net	BII	CHINA
yeti-ns.tsif.net	TSIF	USA
yeti-ns.wide.ad.jp	WIDE Project	Japan
yeti-ns.as59715.net	as59715	Italy
dahu1.yeti.eu.org	Dahu Group	France
ns-yeti.bondis.org	Bond Internet Systems	Spain
yeti-ns.ix.ru	Russia	MSK-IX
yeti.bofh.priv.at	CERT Austria	Austria
yeti.ipv6.ernet.in	ERNET India	India
yeti-dns01.dnsworkshop.org	dnsworkshop /informnis	Germany
dahu2.yeti.eu.org	Dahu Group	France
yeti.aquaray.com	Aqua Ray SAS	France
yeti-ns.switch.ch	SWITCH	Switzerland
yeti-ns.lab.nic.cl	NIC Chile	Chile
yeti-ns1.dns-lab.net	BII	China
yeti-ns2.dns-lab.net	BII	China
yeti-ns3.dns-lab.net	BII	China
ca...a23dc.yeti-dns.net	Yeti-ZA	South Africa
3f...374cd.yeti-dns.net	Yeti-AU	Australia
yeti1.ipv6.ernet.in	ERNET India	India
xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c	ERNET India	India
yeti-dns02.dnsworkshop.org	dnsworkshop /informnis	USA
yeti.mind-dns.nl	Monshouwer Internet Diensten	Netherlands
yeti-ns.datev.net	DATEV	Germany
yeti.jhcloos.net.	jhcloos	USA

The current list of Yeti-Root server is made available to a participating resolver first using a substitute hints file Appendix A and subsequently by the usual resolver priming process [RFC8109]. All Yeti-Root servers are IPv6-only, foreshadowing a future IPv6-only Internet, and hence the Yeti-Root hints file contains no IPv4 addresses and the Yeti-Root zone contains no IPv4 glues. Note that the rationale of IPv6-only testbed is to test whether IPv6-only root can survive or not. Any problem or impact when IPv4 is turn off, much like the context of IETF sunset4 WG (https://datatracker.ietf.org/wg/sunset4/about/).

At the time of writing, all root servers within the Root Server System serve the ROOT-SERVERS.NET zone in addition to the root zone, and all but one also serve the ARPA zone. Yeti-Root servers serve the Yeti-Root zone only.

Significant software diversity exists across the set of Yeti-Root servers, as reported by their volunteer operators at the time of writing:

o Platform: 18 of 25 Yeti-Root servers are implemented on a VPS rather than bare metal.

o Operating System: 15 Yeti-Root servers run on Linux (Ubuntu, Debian, CentOS, Red Hat and ArchLinux); 4 run on FreeBSD, 1 on NetBSD and 1 in Windows server 2016.

o DNS software: 18 of 25 Yeti-Root servers use BIND9 (versions varying between 9.9.7 and 9.10.3); 4 use NSD (4.10 and 4.15); 2 use Knot (2.0.1 and 2.1.0), 1 uses Bundy (1.2.0) and 1 uses MS DNS (10.0.14300.1000).

4.7. Experimental Traffic

For the Yeti DNS Testbed to be useful as a platform for experimentation, it needs to carry statistically representative traffic. Several approaches have been taken to load the system with traffic, including both real-world traffic triggered by end-users and synthetic traffic.

Resolvers that have been explicitly configured to participate in the testbed, as described in Section 4, are a source of real-world, end-user traffic. Due to efficient cache mechanism, the mean query rate is less than 100 qps in Yeti testbed, but a variety of sources are observed active in past one year, as summarized inAppendix C.

Synthetic traffic has been introduced to the system from time to time in order to increase traffic loads. Approaches include the use of distributed measurement platforms such as RIPE ATLAS to send DNS queries to Yeti-Root servers, and the capture of traffic sent from non-Yeti resolvers to the Root Server System which was subsequently modified and replayed towards Yeti-Root servers.

4.8. Traffic Capture and Analysis

Query and response traffic capture is available in the testbed in both Yeti resolvers and Yeti-Root servers in anticipation of experiments that require packet-level visibility into DNS traffic.

Traffic capture is performed on Yeti-Root servers using either dnscap <https://www.dns-oarc.net/tools/dnscap> or pcapdump (part of the pcaputils Debian package <https://packages.debian.org/sid/pcaputils>, with a patch to facilitate triggered file upload <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=545985>. PCAP-format files containing packet captures are uploaded using rsync to central storage.

5. Operational Experience with the Yeti DNS Testbed

The following sections provide commentary on the operation and impact analyses of the Yeti-DNS Testbed described in Section 4. More detailed descriptions of observed phenomena are available in Yeti DNS mailing list archives <http://lists.yeti-dns.org/pipermail/discuss/> and on the Yeti DNS blog <https://yeti-dns.org/blog.html>.

5.1. Viability of IPv6-Only Operation

All Yeti-Root servers were deployed with IPv6 connectivity, and no IPv4 addresses for any Yeti-Root server were made available (e.g. in the Yeti hints file, or in the DNS itself). This implementation decision constrained the Yeti-Root system to be v6-only.

DNS implementations are generally adept at using both IPv4 and IPv6 when both are available. Servers that cannot be reliably reached over one protocol might be better queried over the other, to the benefit of end-users in the common case where DNS resolution is on the critical path for end-users' perception of performance. However, this optimisation also means that systemic problems with one protocol can be masked by the other. By forcing all traffic to be carried over IPv6, the Yeti DNS testbed aimed to expose any such problems and make them easier to identify and understand. Several examples of IPv6-specific phenomena observed during the operation of the testbed are described in the sections that follow.

Although the Yeti-Root servers themselves were only reachable using IPv6, real-world end-users often have no IPv6 connectivity. The testbed was also able to explore the degree to which IPv6-only Yeti-Root servers were able to serve single-stack, IPv4-only end-user populations through the use of dual-stack Yeti resolvers.

5.1.1. IPv6 Fragmentation

In the Root Server System, structural changes with the potential to increase response sizes (and hence fragmentation, fallback to TCP transport or both) have been exercised with great care, since the impact on clients has been difficult to predict or measure. The Yeti DNS Testbed is experimental and has the luxury of a known client

base, making it far easier to make such changes and measure their impact.

Many of the experimental design choices described in this document were expected to trigger larger responses. For example, the choice of naming scheme for Yeti-Root Servers described in Section 4.5 defeats label compression. It makes a large priming response (up to 1754 octets with 25 NS server and their glue) ; the Yeti-Root zone transformation approach described in Section 4.2.2 greatly enlarges the apex DNSKEY RRSet especially during the KSK rollover (up to 1975 octets with 3 ZSK and 2 KSK). An increased incidence of fragmentation was therefore expected.

The Yeti-DNS Testbed provides service on IPv6 only. However middlebox like firewall, and some routers are not friendly on IPv6 fragments. It is reported there is notable packets drop rate due to the mistreatment of middle-box on IPv6 fragment [I-D.taylor-v6ops-fragdrop] [RFC7872]. One APNIC study [IPv6-frag-DNS] reported that 37% of endpoints using IPv6-capable DNS resolver cannot receive a fragmented IPv6 response over UDP.

To study the impact, RIPE Atlas probes were used. For each Yeti-Root server, an Atlas measurement was setup using 100 IPv6-enabled probes from five regions, sending a DNS query for ./IN/DNSKEY using UDP transport with DO=1. This measurement, when carried out concurrently with a Yeti KSK rollover, further exacerbating the potential for fragmentation, identified a 7% failure rate compared with a non-fragmented control. A failure rate of 2% was observed with response sizes of 1414 octets, which was surprising given the expected prevalence of 1500-octet (Ethernet-framed) MTUs.

The consequences of fragmentation were not limited to failures in delivering DNS responses over UDP transport. There were two cases where a Yeti-Root server failed to transfer the Yeti-Root zone from a DM using TCP. DM log files revealed "socket is not connected" errors corresponding to zone transfer requests. Further experimentation revealed that combinations of NetBSD 6.1, NetBSD 7.0RC1, FreeBSD 10.0, Debian 3.2 and VMWare ESXI 5.5 resulted in a high TCP MSS value of 1440 octets being negotiated between client and server despite the presence of the IPV6_USE_MIN_MTU socket option, as described in [I-D.andrews-tcp-and-ipv6-use-minmtu]. The mismatch appears to cause outbound segments greater in size than 1280 octets to be dropped before sending. Setting the local TCP MSS to 1220 octets (chosen as 1280-60, the size of the IPv6/TCP header with no other extension headers) was observed to be a pragmatic mitigation.

5.1.2. Serving IPv4-Only End-Users

Yeti resolvers have been successfully used by real-world end-users for general name resolution within a number of participant organisations, including resolution of names to IPv4 addresses and resolution by IPv4-only end-user devices.

Some participants, recognising the operational importance of reliability in resolver infrastructure and concerned about the stability of their IPv6 connectivity, chose to deploy Yeti resolvers in parallel to conventional resolvers, making both available to end-users. While the viability of this approach provides a useful data point, end-users using Yeti resolvers exclusively provided a better opportunity to identify and understand any failures in the Yeti DNS testbed infrastructure.

Resolvers deployed in IPv4-only environments were able to join the Yeti DNS testbed by way of upstream, dual-stack Yeti resolvers, or in one case, in CERNET2, by assigning IPv4 addresses to Yeti-Root servers and mapping them in dual-stack IVI translation devices [RFC6219].

5.2. Zone Distribution

The Yeti DNS testbed makes use of multiple DMs to distribute the Yeti-Root zone, an approach that would allow the number of Yeti-Root servers to scale to a higher number than could be supported by a single distribution source and which provided redundancy. The use of multiple DMs introduced some operational challenges, however, which are described in the following sections.

5.2.1. Zone Transfers

Yeti-Root Servers were configured to serve the Yeti-Root zone as slaves. Each slave had all DMs configured as masters, providing redundancy in zone synchronisation.

Each DM in the Yeti testbed served a Yeti-Root zone which is functionally equivalent but not congruent to that served by every other DM (see Section 4.3). The differences included variations in the SOA.MNAME field and, more critically, in the RRSIGs for everything other than the apex DNSKEY RRSet, since signatures for all other RRSets are generated using a private key that is only available to the DM serving its particular variant of the zone (see Section 4.2, Section 4.2.2).

Incremental Zone Transfer (IXFR), as described in [RFC1995], is a viable mechanism to use for zone synchronization between any Yeti-

Root server and a consistent, single DM. However, if that Yeti-Root server ever selected a different DM, IXFR would no longer be a safe mechanism; structural changes between the incongruent zones on different DMs would not be included in any transferred delta and the result would be a zone that was not internally self-consistent. For this reason the first transfer after a change of DM would require AXFR, not IXFR.

None of the DNS software in use on Yeti-Root Servers supports this mixture of IXFR/AXFR according to the master server in use. This is unsurprising, given that the environment described above in the Yeti-Root system is idiosyncratic; conventional zone transfer graphs involve zones that are congruent between all nodes. For this reason, all Yeti-Root servers are configured to use AXFR at all times, and never IXFR, to ensure that zones being served are internally self-consistent.

5.2.2. Delays in Yeti-Root Zone Distribution

Each Yeti DM polled the Root Server System for a new revision of the root zone on an interleaved schedule, as described in Section 4.1. Consequently, different DMs were expected to retrieve each revision of the root zone, and make a corresponding revision of the Yeti-Root zone available, at different times. The availability of a new revision of the Yeti-Root zone on the first DM would typically precede that of the last by 40 minutes.

It might be expected given this distribution mechanism that the maximum latency between the publication of a new revision of the root zone and the availability of the corresponding Yeti-Root zone on any Yeti-Root server would be 20 minutes, since in normal operation at least one DM should serve that Yeti-Zone within 20 minutes of root zone publication. In practice, this was not observed.

In one case a Yeti-Root server running Bundy 1.2.0 on FreeBSD 10.2-RELEASE was found to lag root zone publication by as much as ten hours, which upon investigation was due to software defects that were subsequently corrected.

More generally, Yeti-Root servers were observed routinely to lag root zone publication by more than 20 minutes, and relatively often by more than 40 minutes. Whilst in some cases this might be assumed to be a result of connectivity problems, perhaps suppressing the delivery of NOTIFY messages, it was also observed that Yeti-Root servers receiving a NOTIFY from one DM would often send SOA queries and AXFR requests to a different DM. If that DM was not yet serving the new revision of the Yeti-Root zone, a delay in updating the Yeti-Root server would naturally result.

5.2.3. Mixed RRSIGs from different DM ZSKs

The second approach doing the transformation of Root Zone to Yeti-Root Zone (Section 4.2.2) introduce a situation that mixed RRSIGs from different DM ZSKs will be cached in one resolver.

It is observed that the Yeti-Root Zone served by any particular Yeti-Root Server will include signatures generated using the ZSK from the DM that served the Yeti-Root Zone to that Yeti-Root Server. Signatures cached at resolvers might be retrieved from any Yeti-Root Server, and hence are expected to be a mixture of signatures generated by different ZSKs. Since all ZSKs can be trusted through the signature by the Yeti KSK over the DNSKEY RRSet, which includes all ZSKs, the mixture of signatures was predicted not to be a threat to reliable validation.

It was first tested in BII's lab environment as a proof of concept. It is observed in resolver's DNSSEC log that the process of verifying rdataset show "success" with a key (keyid) in DNSKEY RRSet. It was implemented later in three DMs which was carefully coordinated and made public to all Yeti resolver operators and participants in Yeti's mailing list. At least 45 Yeti resolvers (deployed by Yeti operators) were under monitoring and set reporting trigger if anything wrong. In addition, Yeti mailing list is open for error reports from other participants. So far Yeti testbed has been operated in this configuration (with multiple ZSKs) for 2 years. It is proved that this configuration is workable and reliable, even when individual ZSKs are rolled on different schedules.

Another consequence of this approach is that the apex DNSKEY RRSet in the Yeti-Root zone is much larger than the corresponding DNSKEY RRSet in the Root Zone. This requires more space and produce larger response to the query for DNSKEY RRset especially during the KSK rollover.

5.3. DNSSEC KSK Rollover

At the time of writing, the Root Zone KSK is expected to undergo a carefully-orchestrated rollover as described in [ICANN2016]. ICANN has commissioned various tests and has published an external test plan [ICANN2017].

Three related DNSSEC KSK rollover exercises were carried out on the Yeti DNS testbed, somewhat concurrent with the planning and execution of the rollover in the root zone. Brief descriptions of these exercises are included below.

5.3.1. Failure-Case KSK Rollover

The first KSK rollover that was executed on the Yeti DNS testbed deliberately ignored the 30-day hold-down timer specified in [RFC5011] before retiring the outgoing KSK.

It was confirmed that clients of some (but not all) validating Yeti resolvers experienced resolution failures (received SERVFAIL responses) following this change. Those resolvers required administrator intervention to install a functional trust anchor before resolution was restored.

5.3.2. KSK Rollover vs. BIND9 Views

The second Yeti KSK rollover was designed with similar phases to the ICANN's KSK rollover roll, although with modified timings to reduce the time required to complete the process. The "slot" used in this rollover was ten days long, as follows:

	19444	New Key
slot 1	pub+sign	
slot 2,3,4,5	pub+sign	pub
slot 6,7	pub	pub+sign
slot 8	revoke	pub+sign
slot 9		pub+sign

During this rollover exercise, a problem was observed on one Yeti resolver that was running BIND 9.10.4-p2 [KROLL-ISSUE]. That resolver was configured with multiple views serving clients in different subnets at the time that the KSK rollover began. DNSSEC validation failures were observed following the completion of the KSK rollover, triggered by the addition of a new view, intended to serve clients from a new subnet.

BIND 9.10 requires "managed-keys" configuration to be specified in every view, a detail that was apparently not obvious to the operator in this case and which was subsequently highlighted by ISC in their general advice relating to KSK rollover in the root zone to users of BIND 9 <https://www.isc.org/blogs/2017-root-key-rollover-what-does-it-mean-for-bind-users/>. When the "managed-keys" configuration is present in every view that is configured to perform validation, trust anchors for all views are updated during a KSK rollover.

5.3.3. Large Responses during KSK Rollover

Since a KSK rollover necessarily involves the publication of outgoing and incoming public keys simultaneously, an increase in the size of DNSKEY responses is expected. The third KSK rollover carried out on the Yeti DNS testbed was accompanied by a concerted effort to observe response sizes and their impact on end-users.

As described in Section 4.2.2, in the Yeti DNS testbed each DM can maintain control of its own set of ZSKs, which can undergo rollover independently. During a KSK rollover where concurrent ZSK rollovers are executed by each of three DMs the maximum number of apex DNSKEY RRs present is eight (incoming and outcoming KSK, plus incoming and outgoing of each of three ZSKs). In practice, however, such concurrency did not occur; only the BII ZSK was rolled during the KSK rollover, and hence only three DNSKEY RRSet configurations were observed:

- o 3 ZSK and 2 KSK, DNSKEY response of 1975 octets;
- o 3 ZSK and 1 KSK, DNSKEY response of 1414 octets; and
- o 2 ZSK and 1 KSK, DNSKEY response of 1139 octets.

RIPE Atlas probes were used as described in Section 5.1.1 to send DNSKEY queries directly to Yeti-Root servers. The numbers of queries and failures were recorded and categorized according to the response sizes at the time the queries were sent. A summary of the results ([YetiLR]) is as follows:

Response Size	Failures	Total Queries	Failure rate
1139	274	64252	0.0042
1414	3141	126951	0.0247
1975	2920	42529	0.0687

The general approach illustrated briefly here provides a useful example of how the design of the Yeti DNS testbed, separate from the Root Server System but constructed as a live testbed on the Internet, facilitates the use of general-purpose active measurement facilities such as RIPE Atlas probes as well as internal passive measurement such as packet capture.

5.4. Capture of Large DNS Response

Packet capture is a common approach in production DNS systems where operators require fine-grained insight into traffic in order to understand production traffic. For authoritative servers, capture of inbound query traffic is often sufficient, since responses can be synthesised with knowledge of the zones being served at the time the query was received. Queries are generally small enough not to be fragmented, and even with TCP transport are generally packed within a single segment.

The Yeti DNS testbed has different requirements; in particular there is a desire to compare responses obtained from the Yeti infrastructure with those received from the Root Server System in response to a single query stream (e.g. using YmmV as described in Appendix D). Some Yeti-Root servers were capable of recovering complete DNS messages from within nameservers, e.g. using dnstap; however, not all servers provided that functionality and a consistent approach was desirable.

The requirement passive capture of responses from the wire together with experiments that were expected (and in some cases designed) to trigger fragmentation and use of TCP transport led to the development of a new tool, PcapParser, to perform fragment and TCP stream reassembly from raw packet capture data. A brief description of PcapParser is included in Appendix D.

5.5. Automated Hints File Maintenance

Renumbering events in the Root Server System are relatively rare. Although each such event is accompanied by the publication of an updated hints file in standard locations, the task of updating local copies of that file used by DNS resolvers is manual, and the process has an observably-long tail: for example, in 2015 J-Root was still receiving traffic at its old address some thirteen years after renumbering [Wessels2015].

The observed impact of these old, deployed hints file is minimal, likely due to the very low frequency of such renumbering events. Even the oldest of hints file would still contain some accurate root server addresses from which priming responses could be obtained.

By contrast, due to the experimental nature of the system and the fact that it is operated mainly by volunteers, Yeti-Root Servers are added, removed and renumbered with much greater frequency. A tool to facilitate automatic maintenance of hints files was therefore created, [hintUpdate].

The automated procedure followed by the hintUpdate tool is as follows.

1. Use the local resolver to obtain a response to the query ./IN/NS;

2. Use the local resolver to obtain a set of IPv4 and IPv6 addresses for each name server;

3. Validate all signatures obtained from the local resolvers, and confirm that all data is signed;

4. Compare the data obtained to that contained within the currently-active hints file; if there are differences, rotate the old one away and replace it with a new one.

This tool would not function unmodified when used in the Root Server System, since the names of individual Root Servers (e.g. A.ROOT-SERVERS.NET) are not DNSSEC signed. All Yeti-Root Server names are DNSSEC signed, however, and hence this tool functions as expected in that environment.

5.6. Root Label Compression in Knot DNS Server

[RFC1035] specifies that domain names can be compressed when encoded in DNS messages, being represented as one of

1. a sequence of labels ending in a zero octet;

2. a pointer; or

3. a sequence of labels ending with a pointer.

The purpose of this flexibility is to reduce the size of domain names encoded in DNS messages.

It was observed that Yeti-Root Servers running Knot 2.0 would compress the zero-length label (the root domain, often represented as ".") using a pointer to an earlier example. Although legal, this encoding increases the encoded size of the root label from one octet to two; it was also found to break some client software, in particular the Go DNS library. Bug reports were filed against both Knot and the Go DNS library, and both were resolved in subsequent releases.

6. Conclusions

Yeti DNS was designed and implemented as a live DNS root system testbed. It serves a root zone ("Yeti-Root" in this document) derived from the root zone root zone published by the IANA with only those structural modifications necessary to ensure its function in the testbed system. The Yeti DNS testbed has proven to be a useful platform to address many questions that would be challenging to answer using the production Root Server System, such as those included in Section 3.

Indicative findings following from the construction and operation of the Yeti DNS testbed include:

- Operation in a pure IPv6-only environment; confirmation of a significant failure rate in the transmission of large responses (~7%), but no other persistent failures observed. Two cases in which Yeti-Root servers failed to retrieve the Yeti-Root zone due to fragmentation of TCP segments; mitigated by setting a TCP MSS of 1220 octets (see Section 5.1.1).

- Successful operation with three autonomous Yeti-Root zone signers and 25 Yeti-Root servers, and confirmation that IXFR is not an appropriate transfer mechanism of zones that are structurally incongruent across different transfer paths (see Section 5.2).

- ZSK size increased to 2048 bits and multiple KSK rollovers executed to exercise RFC 5011 support in validating resolvers; identification of pitfalls relating to views in BIND9 when configured with "managed-keys" (see Section 5.3).

- Use of natural (non-normalised) names for Yeti-Root servers exposed some differences between implementations in the inclusion of additional-section glue in responses to priming queries; however, despite this inefficiency, Yeti resolvers were observed to function adequately (see Section 4.5).

- It was observed that Knot 2.0 performed label compression on the root (empty) label. This results in an increased encoding size for references to the root label, since a pointer is encoded as two octets whilst the root label itself only requires one (see Section 5.6).

- Some tools were developed in response to the operational experience of running and using the Yeti DNS testbed: DNS fragment and DNS Additional Truncated Response (ATR) for large DNS responses, a BIND9 patch for additional section glue, YmmV and IPv6 defrag for capturing and mirroring traffic. In addition a

tool to facilitate automatic maintenance of hints files was created (see Appendix D).

The Yeti DNS testbed was used only by end-users whose local infrastructure providers had made the conscious decision to do so, as is appropriate for an experimental, non-production system. So far no serious user complains reached Yeti's mailing list during Yeti normal operation. Although adding more instances into Yeti root system may help to better enhance the quality of service, but it is generally accepted that Yeti DNS performance is good enough to serve the purpose of DNS Root testbed.

The experience gained during the operation of the Yeti DNS testbed suggested several topics worthy of further study:

- Priming Truncation and TCP-only Yeti-Root servers: observe and measure the worst-possible case for priming truncation by responding with TC=1 to all priming queries received over UDP transport, forcing clients to retry using TCP. This should also give some insight into the usefulness of TCP-only DNS in general.

- KSK ECDSA Rollover: one possible way to reduce DNSKEY response sizes is to change to an elliptic curve signing algorithm. While in principle this can be done separately for the KSK and the ZSK, the RIPE NCC has done research recently and discovered that some resolvers require that both KSK and ZSK use the same algorithm. This means that an algorithm roll also involves a KSK roll. Performing an algorithm roll at the root would be an interesting challenge.

- Sticky Notify for zone transfer: the non-applicability of IXFR as a zone transfer mechanism in the Yeti DNS testbed could be mitigated by the implementation of a sticky preference for master server for each slave, such that an initial AXFR response could be followed up with IXFR requests without compromising zone integrity in the case (as with Yeti) that equivalent but incongruent versions of a zone are served by different masters.

- Key distribution for zone transfer credentials: the use of a shared secret between slave and master requires key distribution and management whose scaling properties are not ideally suited to systems with large numbers of transfer clients. Other approaches for key distribution and authentication could be considered.

7. Security Considerations

As introduced in Section 4.4,service metadata is synchronized among 3 DMs using Git tool. Any security issue around Git may affect Yeti DM operation. For example, hacker may compromise one DM's git repository and push unwanted changes to Yeti DM system which may introduce a bad root server or bad key for a period of time.

Yeti resolver needs the bootstrapping files to join the testbed, like hint file and trust anchor of Yeti. All required information is published in yeti-dns.org and Github.com. If hacker tampers those websites with a fake page, new resolver may lose its way and configured with a bad root.

8. IANA Considerations

This document requests no action of the IANA.

9. Acknowledgments

The editors would like to acknowledge the contributions of the various and many subscribers to the Yeti DNS Project mailing lists, including the following people who were involved in the implementation and operation of the Yeti DNS testbed itself:

Tomohiro Ishihara, Antonio Prado, Stephane Bortzmeyer, Mickael Jouanne, Pierre Beyssac, Joao Damas, Pavel Khramtsov, Ma Yan, Otmar Lendl, Praveen Misra, Carsten Strotmann, Edwin Gomez, Remi Gacogne, Guillaume de Lafond, Yves Bovard, Hugo Salgado, Li Zhen, Daobiao Gong, Runxia Wan.

The editors also acknowledge the assistance of the Independent Submissions Editorial Board, and of the following reviewers whose opinions helped improve the clarity of this document:

Subramanian Moonesamy, Joe Abley,Paul Mockapetris

10. References

[hintUpdate]
"Hintfile Auto Update", 2015, <https://github.com/BII-Lab/Hintfile-Auto-Update>.

[I-D.andrews-tcp-and-ipv6-use-minmtu]
Andrews, M., "TCP Fails To Respect IPV6_USE_MIN_MTU", draft-andrews-tcp-and-ipv6-use-minmtu-04 (work in progress), October 2015.

[I-D.muks-dns-message-fragments]
Sivaraman, M., Kerr, S., and D. Song, "DNS message fragments", draft-muks-dns-message-fragments-00 (work in progress), July 2015.

[I-D.song-atr-large-resp]
Song, L., "ATR: Additional Truncated Response for Large DNS Response", draft-song-atr-large-resp-00 (work in progress), September 2017.

[I-D.taylor-v6ops-fragdrop]
Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo, M., and T. Taylor, "Why Operators Filter Fragments and What It Implies", draft-taylor-v6ops-fragdrop-02 (work in progress), December 2013.

[ICANN2010]
"DNSSEC Key Management Implementation for the Root Zone", May 2010, <http://www.root-dnssec.org/wp-content/uploads/2010/05/draft-icann-dnssec-keymgmt-01.txt>.

[ICANN2016]
"Root Zone KSK Rollover Plan", 2016, <https://www.iana.org/reports/2016/root-ksk-rollover-design-20160307.pdf>.

[ICANN2017]
"2017 KSK Rollover External Test Plan", July 2016, <https://www.icann.org/en/system/files/files/ksk-rollover-external-test-plan-22jul16-en.pdf>.

[IPv6-frag-DNS]
"Dealing with IPv6 fragmentation in the DNS", August 2017, <https://blog.apnic.net/2017/08/22/dealing-ipv6-fragmentation-dns>.

[ISC-TN-2003-1]
Abley, J., "Hierarchical Anycast for Global Service Distribution", March 2003, <http://ftp.isc.org/isc/pubs/tn/isc-tn-2003-1.txt>.

[ITI2014] "Identifier Technology Innovation Report", May 2014, <https://www.icann.org/en/system/files/files/iti-report-15may14-en.pdf>.

[KROLL-ISSUE]
"A DNSSEC issue during Yeti KSK rollover", 2016, <http://yeti-dns.org/yeti/blog/2016/10/26/A-DNSSEC-issue-during-Yeti-KSK-rollover.html>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, DOI 10.17487/RFC1996, August 1996, <https://www.rfc-editor.org/info/rfc1996>.

[RFC2826] Internet Architecture Board, "IAB Technical Comment on the Unique DNS Root", RFC 2826, DOI 10.17487/RFC2826, May 2000, <https://www.rfc-editor.org/info/rfc2826>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <https://www.rfc-editor.org/info/rfc2845>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <https://www.rfc-editor.org/info/rfc5011>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, DOI 10.17487/RFC6219, May 2011, <https://www.rfc-editor.org/info/rfc6219>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[RFC7720] Blanchet, M. and L-J. Liman, "DNS Root Name Service Protocol and Deployment Requirements", BCP 40, RFC 7720, DOI 10.17487/RFC7720, December 2015, <https://www.rfc-editor.org/info/rfc7720>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[RFC8109] Koch, P., Larson, M., and P. Hoffman, "Initializing a DNS Resolver with Priming Queries", BCP 209, RFC 8109, DOI 10.17487/RFC8109, March 2017, <https://www.rfc-editor.org/info/rfc8109>.

[RRL] Vixie, P. and V. Schryver, "Response Rate Limiting (RRL)", June 2012, <http://www.redbarn.org/dns/ratelimits>.

[RSSAC001] "Service Expectations of Root Servers", December 2015, <https://www.icann.org/en/system/files/files/rssac-001-root-service-expectations-04dec15-en.pdf>.

[RSSAC023] "History of the Root Server System", November 2016, <https://www.icann.org/en/system/files/files/rssac-023-04nov16-en.pdf>.

[TNO2009] Gijsen, B., Jamakovic, A., and F. Roijers, "Root Scaling Study: Description of the DNS Root Scaling Model", September 2009, <https://www.icann.org/en/system/files/files/root-scaling-model-description-29sep09-en.pdf>.

[Wessels2015] Wessels, D., "Thirteen Years of Old J-Root", 2015, <https://indico.dns-oarc.net/event/24/session/10/contribution/10/material/slides/0.pdf>.

[YetiLR] "Observation on Large response issue during Yeti KSK rollover", August 2017, <http://yeti-dns.org/yeti/blog/2017/08/02/large-packet-impact-during-yeti-ksk-rollover.html>.

Appendix A. Yeti-Root Hints File

The following hints file (complete and accurate at the time of writing) causes a DNS resolver to use the Yeti DNS testbed in place of the production Root Server System and hence participate in experiments running on the testbed.

Note that some lines have been wrapped in the text that follows in order to fit within the production constraints of this document. Wrapped lines are indicated with a blackslash character ("\"), following common convention.

```
.                      3600000  IN   NS     bii.dns-lab.net
bii.dns-lab.net        3600000  IN   AAAA   240c:f:1:22::6
.                      3600000  IN   NS     yeti-ns.tisf.net
yeti-ns.tisf.net       3600000  IN   AAAA   2001:559:8000::6
.                      3600000  IN   NS     yeti-ns.wide.ad.jp
yeti-ns.wide.ad.jp     3600000  IN   AAAA   2001:200:1d9::35
.                      3600000  IN   NS     yeti-ns.as59715.net
yeti-ns.as59715.net    3600000  IN   AAAA   \
                            2a02:cdc5:9715:0:185:5:203:53
.                      3600000  IN   NS     dahu1.yeti.eu.org
dahu1.yeti.eu.org      3600000  IN   AAAA   \
                            2001:4b98:dc2:45:216:3eff:fe4b:8c5b
.                      3600000  IN   NS     ns-yeti.bondis.org
ns-yeti.bondis.org     3600000  IN   AAAA   2a02:2810:0:405::250
.                      3600000  IN   NS     yeti-ns.ix.ru
yeti-ns.ix.ru          3600000  IN   AAAA   2001:6d0:6d06::53
.                      3600000  IN   NS     yeti.bofh.priv.at
yeti.bofh.priv.at      3600000  IN   AAAA   2a01:4f8:161:6106:1::10
.                      3600000  IN   NS     yeti.ipv6.ernet.in
yeti.ipv6.ernet.in     3600000  IN   AAAA   2001:e30:1c1e:1::333
.                      3600000  IN   NS     yeti-dns01.dnsworkshop.org
yeti-dns01.dnsworkshop.org \
                       3600000  IN   AAAA   2001:1608:10:167:32e::53
.                      3600000  IN   NS     yeti-ns.conit.co
yeti-ns.conit.co       3600000  IN   AAAA   \
                           2604:6600:2000:11::4854:a010
.                      3600000  IN   NS     dahu2.yeti.eu.org
dahu2.yeti.eu.org      3600000  IN   AAAA   2001:67c:217c:6::2
.                      3600000  IN   NS     yeti.aquaray.com
yeti.aquaray.com       3600000  IN   AAAA   2a02:ec0:200::1
```

```
   .                      3600000  IN   NS     yeti-ns.switch.ch
   yeti-ns.switch.ch      3600000  IN   AAAA   2001:620:0:ff::29
   .                      3600000  IN   NS     yeti-ns.lab.nic.cl
   yeti-ns.lab.nic.cl     3600000  IN   AAAA   2001:1398:1:21::8001
   .                      3600000  IN   NS     yeti-ns1.dns-lab.net
   yeti-ns1.dns-lab.net   3600000  IN   AAAA   2001:da8:a3:a027::6
   .                      3600000  IN   NS     yeti-ns2.dns-lab.net
   yeti-ns2.dns-lab.net   3600000  IN   AAAA   2001:da8:268:4200::6
   .                      3600000  IN   NS     yeti-ns3.dns-lab.net
   yeti-ns3.dns-lab.net   3600000  IN   AAAA   2400:a980:30ff::6
   .                      3600000  IN   NS     \
                            ca978112ca1bbdcafac231b39a23dc.yeti-dns.net
   ca978112ca1bbdcafac231b39a23dc.yeti-dns.net \
                          3600000  IN   AAAA   2c0f:f530::6
   .                      3600000  IN   NS     \
                            3e23e8160039594a33894f6564e1b1.yeti-dns.net
   3e23e8160039594a33894f6564e1b1.yeti-dns.net \
                          3600000  IN   AAAA   2803:80:1004:63::1
   .                      3600000  IN   NS     \
                            3f79bb7b435b05321651daefd374cd.yeti-dns.net
   3f79bb7b435b05321651daefd374cd.yeti-dns.net \
                          3600000  IN   AAAA   2401:c900:1401:3b:c::6
   .                      3600000  IN   NS     \
                            xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c
   xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c \
                          3600000  IN   AAAA   2001:e30:1c1e:10::333
   .                      3600000  IN   NS     yeti1.ipv6.ernet.in
   yeti1.ipv6.ernet.in    3600000  IN   AAAA   2001:e30:187d::333
   .                      3600000  IN   NS     yeti-dns02.dnsworkshop.org
   yeti-dns02.dnsworkshop.org \
                          3600000  IN   AAAA   2001:19f0:0:1133::53
   .                      3600000  IN   NS     yeti.mind-dns.nl
   yeti.mind-dns.nl       3600000  IN   AAAA   2a02:990:100:b01::53:0
```

Appendix B. Yeti-Root Server Priming Response

Here is the reply of a Yeti root name server to a priming request. The authoritative server runs NSD.

```
   ...
   ;; Got answer:
   ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62391
   ;; flags: qr aa rd; QUERY: 1, ANSWER: 26, AUTHORITY: 0, ADDITIONAL: 7
   ;; WARNING: recursion requested but not available

   ;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags: do; udp: 1460
;; QUESTION SECTION:
;.                      IN NS

;; ANSWER SECTION:
.            86400 IN NS bii.dns-lab.net.
.            86400 IN NS yeti.bofh.priv.at.
.            86400 IN NS yeti.ipv6.ernet.in.
.            86400 IN NS yeti.aquaray.com.
.            86400 IN NS yeti.jhcloos.net.
.            86400 IN NS yeti.mind-dns.nl.
.            86400 IN NS dahu1.yeti.eu.org.
.            86400 IN NS dahu2.yeti.eu.org.
.            86400 IN NS yeti1.ipv6.ernet.in.
.            86400 IN NS ns-yeti.bondis.org.
.            86400 IN NS yeti-ns.ix.ru.
.            86400 IN NS yeti-ns.lab.nic.cl.
.            86400 IN NS yeti-ns.tisf.net.
.            86400 IN NS yeti-ns.wide.ad.jp.
.            86400 IN NS yeti-ns.datev.net.
.            86400 IN NS yeti-ns.switch.ch.
.            86400 IN NS yeti-ns.as59715.net.
.            86400 IN NS yeti-ns1.dns-lab.net.
.            86400 IN NS yeti-ns2.dns-lab.net.
.            86400 IN NS yeti-ns3.dns-lab.net.
.            86400 IN NS xn--r2bi1c.xn--h2bv6c0a.xn--h2brj9c.
.            86400 IN NS yeti-dns01.dnsworkshop.org.
.            86400 IN NS yeti-dns02.dnsworkshop.org.
.            86400 IN NS 3f79bb7b435b05321651daefd374cd.yeti-dns.net.
.            86400 IN NS ca978112ca1bbdcafac231b39a23dc.yeti-dns.net.
.            86400 IN RRSIG NS 8 0 86400 (
                        20171121050105 20171114050105 26253 .
                        FUvezvZgKtlLzQx2WKyg+D6dw/pITcbuZhzStZfg+LNa
                        DjLJ9oGIBTU1BuqTujKHdxQn0DcdFh9QE68EPs+93bZr
                        VlplkmObj8f0B7zTQgGWBkI/K4Tn6bZ1I7QJ0Zwnk1mS
                        BmEPkWmvo0kkaTQbcID+tMTodL6wPAgW1AdwQUInfy21
                        p+31GGm3+SU6SJsgeHOzPUQW+dUVWmdj6uvWCnUkzW9p
                        +5en4+85jBfEOf+qiyvaQwUUe98xZ1TOiSwYvk5s/qiv
                        AMjG6nY+xndwJUwhcJAXBVmGgrtbiR8GiGZfGqt748VX
                        4esLNtD8vdypucffem6n0T0eV1c+7j/eIA== )

;; ADDITIONAL SECTION:
bii.dns-lab.net.        86400 IN AAAA 240c:f:1:22::6
yeti.bofh.priv.at.      86400 IN AAAA 2a01:4f8:161:6106:1::10
yeti.ipv6.ernet.in.     86400 IN AAAA 2001:e30:1c1e:1::333
yeti.aquaray.com.       86400 IN AAAA 2a02:ec0:200::1
yeti.jhcloos.net.       86400 IN AAAA 2001:19f0:5401:1c3::53
yeti.mind-dns.nl.       86400 IN AAAA 2a02:990:100:b01::53:0
```

```
;; Query time: 163 msec
;; SERVER: 2001:4b98:dc2:45:216:3eff:fe4b:8c5b#53
;; WHEN: Tue Nov 14 16:45:37 +08 2017
;; MSG SIZE  rcvd: 1222
```

Appendix C. Active IPv6 Prefixes in Yeti DNS testbed

Prefix	Originator	Location
240c::/28	BII	CN
2001:6d0:6d06::/48	MSK-IX	RU
2001:1488::/32	CZ.NIC	CZ
2001:620::/32	SWITCH	CH
2001:470::/32	Hurricane Electric, Inc.	US
2001:0DA8:0202::/48	BUPT6-CERNET2	CN
2001:19f0:6c00::/38	Choopa, LLC	US
2001:da8:205::/48	BJTU6-CERNET2	CN
2001:62a::/31	Vienna University Computer Center	AT
2001:67c:217c::/48	AFNIC	FR
2a02:2478::/32	Profitbricks GmbH	DE
2001:1398:1::/48	NIC Chile	CL
2001:4490:dc4c::/46	NIB (National Internet Backbone)	IN
2001:4b98::/32	Gandi	FR
2a02:aa8:0:2000::/52	T-Systems-Eltec	ES
2a03:b240::/32	Netskin GmbH	CH
2801:1a0::/42	Universidad de Ibague	CO
2a00:1cc8::/40	ICT Valle Umbra s.r.l.	IT
2a02:cdc0::/29	ORG-CdSB1-RIPE	IT

Appendix D. Tools developed for Yeti DNS testbed

Various tools were developed to support the Yeti DNS testbed, a selection of which are described briefly below.

YmmV ("Yeti Many Mirror Verifier") is designed to make it easy and safe for a DNS administrator to capture traffic sent from a resolver to the Root Server System and to replay it towards Yeti-Root servers. Responses from both systems are recorded and compared, and differences are logged. See <https://github.com/BII-Lab/ymmv>.

PcapParser is a module used by YmmV which reassembles fragmented IPv6 datagrams and TCP segments from a PCAP archive and extracts DNS

messages contained within them. See <https://github.com/RunxiaWan/PcapParser>.

DNS-layer-fragmentation implements DNS proxies that perform application-level fragmentation of DNS messages, based on [I-D.muks-dns-message-fragments]. The idea with these proxies is to explore splitting DNS messages in the protocol itself, so they will not by fragmented by the IP layer. See <https://github.com/BII-Lab/DNS-layer-Fragmentation>.

DNS_ATR is an implementation of DNS Additional Truncated Response (ATR), as described in [I-D.song-atr-large-resp]. DNS_ATR acts as a proxy between resolver and authoritative servers, forwarding queries and responses as a silent and transparent listener. Responses that are larger than a nominated threshold (1280 octets by default) trigger additional truncated responses to be sent immediately following the large response. See <https://github.com/songlinjian/DNS_ATR>.

Appendix E. Controversy

The Yeti DNS Project, its infrastructure and the various experiments that have been carried out using that infrastructure, have been described by people involved in the project in many public meetings at technical venues since its inception. The mailing lists using which the operation of the infrastructure has been coordinated are open to join, and their archives are public. The project as a whole has been the subject of robust public discussion.

Some commentators have expressed concern that the Yeti DNS Project is, in effect, operating an alternate root, challenging the IAB's comments published in [RFC2826]. Other such alternate roots are considered to have caused end-user confusion and instability in the namespace of the DNS by the introduction of new top-level labels or the different use of top-level labels present in the Root Server System. The coordinators of the Yeti DNS Project do not consider the Yeti DNS Project to be an alternate root in this sense, since by design the namespace enabled by the Yeti-Root Zone is identical to that of the Root Zone.

Some commentators have expressed concern that the Yeti DNS Project seeks to influence or subvert administrative policy relating to the Root Server System, in particular in the use of DNSSEC trust anchors not published by the IANA and the use of Yeti-Root Servers in regions where governments or other organisations have expressed interest in operating a Root Server. The coordinators of the Yeti-Root project observe that their mandate is entirely technical and has no ambition to influence policy directly; they do hope, however, that technical

findings from the Yeti DNS Project might act as a useful resource for the wider technical community.

Appendix F. About This Document

This section (and sub-sections) has been included as an aid to reviewers of this document, and should be removed prior to publication.

F.1. Venue

The authors propose that this document proceed as an Independent Submission, since it documents work that, although relevant to the IETF, has been carried out externally to any IETF working group. However, a suitable venue for discussion of this document is the dnsop working group.

Information about the Yeti DNS project and discussion relating to particular experiments described in this document can be found at <https://yeti-dns.org/>.

This document is maintained in GitHub at <https://github.com/BII-Lab/yeti-testbed-experience>.

F.2. Revision History

F.2.1. draft-song-yeti-testbed-experience-00 through -03

Change history is available in the public GitHub repository where this document is maintained: <https://github.com/BII-Lab/yeti-testbed-experience>.

F.2.2. draft-song-yeti-testbed-experience-04

Substantial editorial review and rearrangement of text by Joe Abley at request of BII.

Added what is intended to be a balanced assessment of the controversy that has arisen around the Yeti DNS Project, at the request of the Independent Submissions Editorial Board.

Changed the focus of the document from the description of individual experiments on a Root-like testbed to the construction and motivations of the testbed itself, since that better describes the output of the Yeti DNS Project to date. In the considered opinion of this reviewer, the novel approaches taken in the construction of the testbed infrastructure and the technical challenges met in doing so are useful to record, and the RFC series is a reasonable place to

record operational experiences related to core Internet infrastructure.

Note that due to draft cut-off deadlines some of the technical details described in this revision of the document may not exactly match operational reality; however, this revision provides an indicative level of detail, focus and flow which it is hoped will be helpful to reviewers.

F.2.3. draft-song-yeti-testbed-experience-05

Added commentary on IPv6-only operation, IPv6 fragmentation, applicability to and use by IPv4-only end-users and use of multiple signers.

F.2.4. draft-song-yeti-testbed-experience-06

Conclusion; tools; editorial changes.

Authors' Addresses

Linjian Song (editor)
Beijing Internet Institute
2508 Room, 25th Floor, Tower A, Time Fortune
Beijing 100028
P. R. China

Email: songlinjian@gmail.com
URI: http://www.biigroup.com/

Dong Liu
Beijing Internet Institute
2508 Room, 25th Floor, Tower A, Time Fortune
Beijing 100028
P. R. China

Email: dliu@biigroup.com
URI: http://www.biigroup.com/

Paul Vixie
TISF
11400 La Honda Road
Woodside, California 94062
US

Email: vixie@tisf.net
URI: http://www.redbarn.org/

Akira Kato
Keio University/WIDE Project
Graduate School of Media Design, 4-1-1 Hiyoshi, Kohoku
Yokohama 223-8526
JAPAN

Email: kato@wide.ad.jp
URI: http://www.kmd.keio.ac.jp/

Shane Kerr
Antoon Coolenlaan 41
Uithoorn 1422 GN
NL

Email: shane@time-travellers.org