CONNECT TO

PORT 53

DNS HACKATHON 2023

PORT 53
DNS HACKATHON 2023

DNSSEC Bootstrapping

Peter Thomassen, Joeri de Ruiter, Vincent Jumpertz, Rudi Kraus, Adam Burns

SustainabiliTeam

Lain Aleix Samael Högy, Alex Pion, Malick Koupam Alassane, Joseph Gersch, Vesna Manojlovic

DNS Today + DNS Home Appliance

Jonathan Magnusson, Artyom Minasyan, Kurt Kayser, Sarder Kamal, Lavie B.B., Q Misell, James Rice, Bernhard Degen

DNS and the application layer - aka DApper

Yuang Yuan, Asbjoern Sloth Toennesen, Jelmer Vernooij, Niall

DIG-alicious

Tony Finch, Sebastian Castro, Raffaele Sommese, Jonas Andersson, Alex Semenyaka, Annika Hannig, Emile Aben

DNS Out of protocol signaling - aka DNS Oops

Doris Hauser, Willem Toorop, Henrik Kramselund, Stefan Ubbink, Vilhelm Prytz, Lars-Johan Liman

0 DAYS SINCE IT WAS BGP

(Whoa, it wasn't DNS?)

DNS OOPS

Notify the BGP daemons

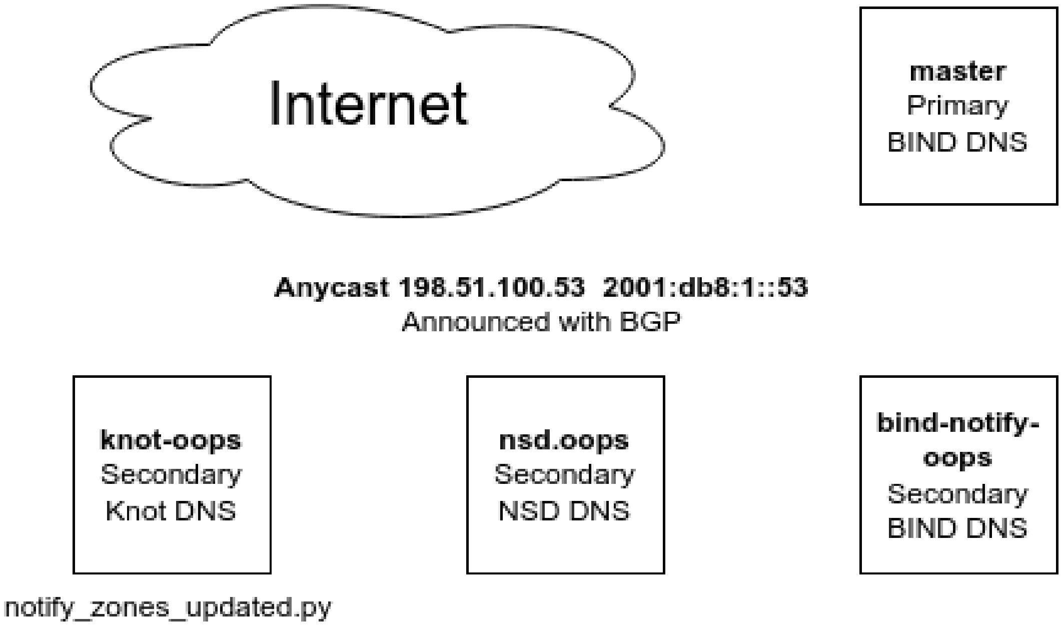
by Willem Toorop (NLnetLabs) Doris Hauser (nic.at) Lars-Johan Liman (netnod)
Vilhelm Prytz (netnod) Stefan Ubbink (SIDN) Henrik Kramselund (Zencurity)

Goals for the project

This document seeks to specify **a method for name servers** to **signal programs outside** of the **name server software**, and which are not necessarily involved with the DNS protocol, **about conditions** that can arise **within the name server**. These **signals** can be used to **invoke actions** in areas that **help provide the DNS service**, such as **routing**.

Main URL:
https://datatracker.ietf.org/doc/draft-grubto-dnsop-dns-out-of-protocol-signalling/

The goal



Internet

master
Primary
BIND DNS

Anycast 198.51.100.53 2001:db8:1::53
Announced with BGP

knot-oops
Secondary
Knot DNS

notify_zones_updated.py

nsd.oops
Secondary
NSD DNS

bind-notify-oops
Secondary
BIND DNS

Automated signalling when things happen in the DNS server ...

Technologies used

- DNS server software: Knot DNS, BIND, NSD
- BGP software: Bird, ExaBGP
- Programming languages: Python 3, Shell scripting
- Notification: shell scripting, D-bus to Knot DNS

Example: Use the Libraries

First Python script uses various features:

```
#!/usr/bin/env python3

import sys
import dbus    // D-bus is implemented in Knot DNS
import dbus.mainloop.glib
import signal
import requests
import json
import libknot.control
from gi.repository import GLib

api_url = None
node_data_file = '/root/node_data.json'
zones = {}
```

Example: Event loop and D-bus

When zone is updated, do something:

```
def updated(*args, **kwargs):
    """
    Event handler for Glib.MainLoop
    Also explicitly called at script startup to determine if any event handling needs to be done
    """
    (zone, serial) = args
    print("Zone %s updated, SOA serial %d" % (zone, serial))
    zones[zone] = serial
    if None not in zones.values():
        # Build a list of all zones with serials
        all_zone_info = ""
        for key in zones:
            all_zone_info += f'{key}:{zones[key]};'
...
```

This is an example, can be run inside/outside of the server, in cloud systems etc.

Improvements and next steps

- Draft being updated https://datatracker.ietf.org/doc/draft-grubto-dnsop-dns-out-of-protocol-signalling/
- Create dockerfiles for packing this in an easy way
- Add some bridging between Notify and D-bus
- Add D-bus features to NSD
- NSD issue and branch being created https://github.com/NLnetLabs/nsd/pull/280