Test cases for domain checks – a step towards a best practice

Mats Dufberg, .SE

Sandoche Balakrichenan, AFNIC

Zonemaster

- Upcoming tool for test of delegation of a domain
- The development of Zonemaster has several purposes
    - Replace tools built by Afnic (Zonecheck) and .SE (DNSCheck), respectively with a better tool.
    - Create a tool that should be modular and easy to update.
    - Create explicit requirements for correct delegation of a domain.
    - Create explicit requirements that can be used to verify the tool

The goals of the requirements

- General accepted requirements of a delegation of a domain.
- Requirements accepted as a best practice for a domain.
- Requirements for domain and tool that could be used for any tool, not only Zonemaster.

Requirements on tool and specification for domain delegation

- Requirements on tool:
  - The tool must be able to detect errors and certain behavior to make sure that we do not get false positives or false negatives.
- Specification for delegation of a domain:
  - These are everything that we require on the domain to be considered to correctly delegated.

Why this presentation?

- Is this right approach?
- Is it possible to define generally accepted requirements? Can we reach a best practice for testing the delegation of a domain?
- Can we reach the point where different tools test the same thing? – If two come to different conclusions, one is probably wrong?

- What are your thoughts? Have we missed something?

Requirements on the requirements

- The requirements must be SMaRT
  - Specific – It must be clear what the requirement is.
  - Measurable – It must be possible to issue a DNS query, or a set of queries, and from that determine if the requirement has been fulfilled or not.
  - Realistic – We cannot assume more access to the name servers than we normally have.
  - Time – It cannot be too time-consuming to perform the tests.
- In this presentation all details are, of course, not included. See https://github.com/dotse/zonemaster for more details.

Requirements on tool

- The tool must be able to detect restrictions, such as hostname syntax, and configuration errors, such as CNAME collision and lame delegation.
- The tool must be able to differentiate between different status of an answer, such as NXDOMAIN and psuedo-status NODATA.
- This is work in progress. More requirements to come.

Explicit specification for the delegation of a domain

- The specifications fall into groups:
  - Basic
  - Delegation
  - Address
  - Connectivity
  - Consistency
  - Name server
  - Syntax
  - Zone
  - DNSSEC

Source of specifications

- The specifications are, as long as it is possible, based on documented requirements, mostly RFCs.
- Some specifications are based on best practice and experience of DNS operation.
- In all cases, the requirements are stated at the Zonemaster Github site, http://goo.gl/Z4yxTR (https://github.com/dotse/zonemaster to main site).

Basic

- The domain must meet the following requirements, or else the rest of the requirements are meaningless:
  - The domain must have a parent domain
  - The domain must have at least one working name server

Delegation

- The delegation is here seen as the overlap between the delegating zone and the zone in question.
  - At least two NS
  - Each NS must resolve to a distinct address
  - Referral must fit 512 bytes
  - NS must be authoritative
  - NS must not point at CNAME
  - SOA must exist
  - Glue in delegation must exist in zone

Address

- The resolved IP addresses must meet some requirements:
  - Address must be globally routable
  - No addresses in bogon prefixes
  - Address should have reverse (PTR)
  - Reverse (PTR) should match the name

Connectivity

- Connectivity is fundamental.
  - UDP
  - TCP
  - AS and network diversity

Consistency

- Consistency between data from different name servers of the same domain is needed for a healthy domain.
  - All parameters in SOA equal between all server.
  - The same NS RR set in all servers.

Name server

- The behavior of the name servers used for the domain will be fundamental to the quality of the domain.
  - Must handle queries for AAAA correctly.
  - Must respond from the same IP as the query came to.
  - Should or must support EDNS0
  - Should not be a recursor.
  - Should not allow AXFR from ”the world”.

Syntax

- The format of names must meet the standards.
  - No illegal characters in domain name (only LDH)
  - No hyphen in initial or final position of any label.
  - "xn--" is OK, but else "--" is not OK in third and forth position
  - Name server name must be valid
  - No "@" in SOA RNAME (mail addr)
  - No illegal characters in SOA RNAME (mail addr)
  - No illegal characters in SOA MNAME (master server)
  - Any MX in apex must point at a valid hostname

Zone

- A complete analysis of the zone would require AXFR. Some parts can be tested.
  - Name server in SOA MNAME must be a fully qualified master server for the zone.
  - SOA values (refresh, retry, expire, minimum) must be sensible
  - SOA MNAME must not point at a CNAME
  - The zone should have an MX in apex (unless it is the root or a TLD)
  - MX must not point at a CNAME
  - The zone should have an A or AAAA in apex unless it has a valid MX (unless it is the root or a TLD)

DNSSEC

- The DNSSEC tests are of course only relevant for signed zones.
  - If it has a DS record in the delegation, it must be signed
  - Legal values for DS hash digest algorithms
  - DS must match DNSKEY in the zone
  - Not too many NSEC3 iterations
  - Not too short or too long RRSIG lifetime
  - DNSKEY algorithm must be valid
  - Answers must contain RRISG RR
  - Signed zone should have a DS in parent
  - RRSIG of DNSKEY and SOA must be valid
  - Zone must have NSEC or NSEC3

A framework

- Instead of focusing on the tool doing the cool testing we have here presented the requirements.
- If we succeed, the requirements will turn into a best practice and living outside the Zonemaster tool.

Thank you.

Questions and comments?