Board Question #2

The role that negative answers currently returned from queries to the root for these strings play in the experience of the end user, including in the operation of existing end systems.

As noted in the SSAC Report, "Redirection in the Com and Net Domains," uninstantiated names that result in negative answers might occur for a variety of reasons: "A name might not exist because it had been misspelled, had lapsed or had never been registered. A name might also be registered or reserved but not included in the lookup database used for domain name queries."[1] Regardless of the reason, the errors received when returning a negative answer are in and of themselves useful to systems and end users. For example, systems such as spam filtering services may rely on the error to help determine if a message is spam by checking whether the domain name of the sender exists.

Any interruption or intervention in the path that results in a negative answer has the potential to intrude upon end-user privacy by allowing the intervening system to collect data on the user's behavior and the path attempted.[2] From a system perspective, interruption or intervention in the flow by a third party could result in increased network charges for some classes of users, a reduction in performance, or the creation of work required to compensate for the consequent failure.[3]

Workspace: for data and documentation

Applicable notes from Study 1

* Ke SSAC siteFinder report likely has all the information we need to respond to this question
* https://www.icann.org/en/system/files/files/report-redirection-com-net-09jul04-en.pdf

- [Why might there be a negative response] Names might be uninstantiated for one of several reasons: A name might not exist because it had been misspelled, had lapsed or had never been

---

[1] pg 3
[2] pg 22
[3] pg 23

registered. A name might also be registered or reserved but not included in the lookup database used for domain name queries. An error is a legitimate form of information.

- Users' decisions and control were preempted and users were potentially subjected to violations of their privacy.
    - Information about intended e-mail senders and receivers was necessarily accepted by VeriSign's servers without the knowledge or consent of either sender or receiver.
    - The behavior of end users redirected to the Web site was observed by a program embedded in the Site Finder service, and users could neither accept it, reject it nor substitute another, similar service for it
- certain e-mail systems, spam filters and other services failed resulting in direct and indirect costs to third parties, either in the form of increased network charges for some classes of users, a reduction in performance, or the creation of work required to compensate for the consequent failure.
    - One of the strategies used by some spam filters is to check whether the domain name of the sender exists.
- Most Web and e-mail end users have seen error messages when a name fails to resolve. These error messages usually come either as a Web page displayed on their browsers, perhaps supported by a well-known search service, or as a bounced message in their e-mail in-boxes

* NSEC caching and QNAME minimization effects the visibility of NXDOMAIN in root data
* other technology changes that have effected visibility of names at root servers
Proposed Gap:
* Will need to add discussion about NSEC caching and QNAME minimization because not covered by SiteFinder
* other technology changes discussions (localroot)
* consider queries of existing root data (and resolver data?)

Questions from Study 2 Proposal, Appendix 3

- How has application logic evolved to depend on DNS (while being cognizant many legacy systems that are not well understood still persist)?
- Are there examples of new technologies that take advantage of NXDOMAIN?
- Can specific systems or trends be identified by looking into the data to find new software that relies on non-delegated strings?
- Why are people and systems still explicitly relying on non-delegated strings?
- What advice can be given to people so that maybe they'll behave better?

RE: the explicit dependency - the bifurcation of the stub resolvers into different app stacks has changed since 2012. That's going to have an impact going forward, and may limit the value of the information in the earlier report and research. Application logic may change based on the DNS responses.

- What is the role of the addition of applications incorporating stub resolvers directly rather than depending on recursive/iterative resolvers