DNS and the Fate of Successful Protocols

"No one goes there any more, it's too crowded" (with apologies to Yogi Berra)

Suzanne Woolf, Warren Kumari

ICANN DNS Symposium, July 13, 2018

Is DNS a Successful Protocol?

Introducing RFC 5218, “What Makes for a Successful Protocol?”

- Basic Success: a protocol fits its intended place in the world; it solves the problem intended at the scale intended. Some helpful factors:
  - Clear problem to be solved
  - Open source
  - Open specification
  - Incremental deployability
- Wild Success: a protocol exceeds its intended uses, scale, or both
  - Extensible
  - No hard scaling limit
  - Threats sufficiently mitigated

What’s after “Wild success”?

- Design decisions may be appropriate for the originally intended purpose but inappropriate for new purposes.
- There may be performance problems if the protocol was not designed to scale to the extent to which it was deployed.
- Implementers may attempt to add or change functionality to work around the design limitations without complete understanding of their effect on the overall protocol behavior and invariants.
- Popularity and the potential for exploitation of poorly-implemented or poorly-understood extensions make wildly successful protocols into high value targets for attackers.

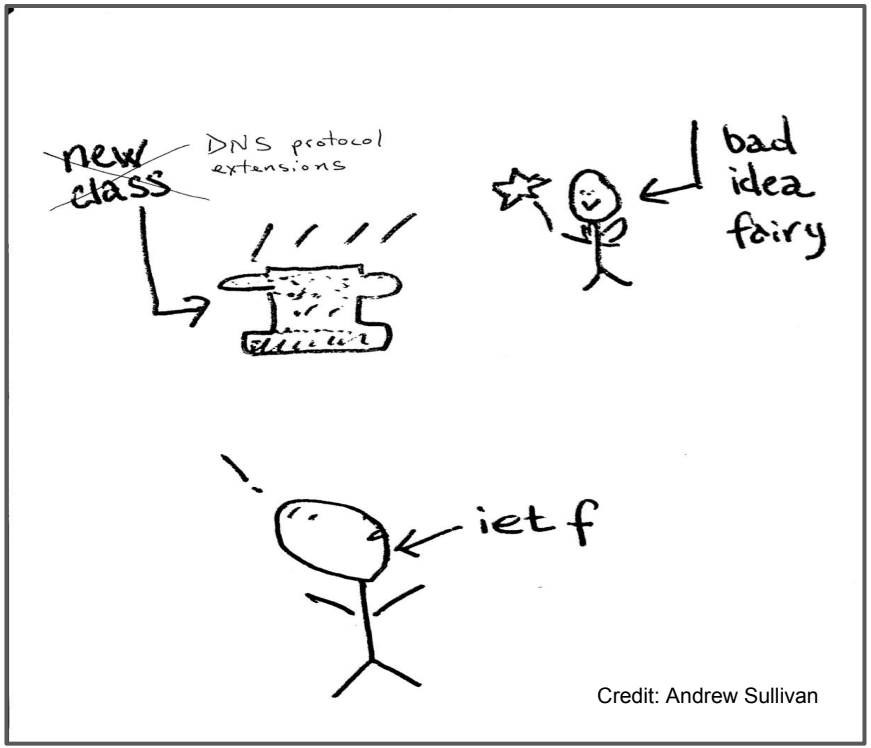
Sound familiar??

- For almost as long as we've had DNS, we've been predicting the imminent "death by success" of DNS
- Arguably it gets more
  - Complex
  - Difficult to implement and operate
  - Difficult to secure
  - ... every year
- **It refuses to die**
- **And we keep talking about it**
  - Bert Hubert, https://datatracker.ietf.org/meeting/101/materials/slides-101-dnsop-sessa-the-dns-camel-01
  - John Klensin, RFC 8324 ("DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look?")

Applying the model: does it work for DNS?

- It seems obvious that:
  - Some evolutionary changes are more robust and more useful than others
  - Even robust, useful changes can interact with other robust, useful changes to create complexity, ambiguity, and an expanded attack surface
- It also seems likely that standards play a part in protocol maintenance
  - A community-driven standards process provides more review of new protocol elements than a single company or open source group: pool clue, promote good effects, mitigate harmful ones
  - **Interoperability is what makes it “the internet”**
- So how do standards help the balance between “wild success” and “death by success”?

About DNS and the IETF Standards Process….



This will look familiar to DNSOP regulars.

Everybody laughed….

Because it’s true….

About the protocol extension proposed at the time….And many before and since.

But it’s easy to be glib, and hard to be discerning.

Apply the analysis to DNS

- Interoperation of features
  - DNSSEC makes a lot of things harder
  - Aliasing (CNAME as specified? DNAME? ANAME? Allow CNAME-at-apex after all?)
- Dueling use cases
  - Enterprise scope vs. global scope; public vs. private
  - Conscious, deliberate use vs. lowest common denominator
- DNS has never had a core specification
  - “Protocol police” is not a thing
  - Lots of informational RFCs
  - Early standards-track RFCs are not rigorous by comparison to modern ones
- Operators, implementers, and the cycle of complexity
- Does interoperation fall by the wayside?

And another question: the Robustness Principle

- The Robustness Principle (“Be strict in what you send…”) is often invoked when people argue about whether to implement, or standardize, or use a particular feature.
- It was intended as a way to speed up the cycle of feature development, feature use, and feature standardization in new protocols
- But does it also have some blame for the decay of utility in a wildly successful protocol? (“Harmful Consequences of the Robustness Principle”, draft-iab-protocol-maintenance-00)
  - Use cases multiply
  - Edges around what a protocol does or does not do get blurred

Role of Standards

- DNSOP has opted recently for “document it if people are doing it”
  - Previous generations of DNS WGs opted to be gatekeepers; people extended DNS anyway, they just didn’t have RFCs about it
  - This obviously promotes interoperability -- or does it? Implementers are drowning in documents and features, and operators are watching attack vectors multiply as RFCs do
- Backlash: “if the IETF produces fewer documents, there will be less pressure on implementers to write goofy new features operators can’t use”
  - This assumes that the standard drives implementation
  - This obviously promotes interoperability-- or does it? We’ve already got myriads of only-marginally-compatible proprietary features littering the operational world; denying them RFCs doesn’t prevent them from being written or used
- Declare victory and go home? DNS-NG? Close DNSOP? Keep trying?

Current DNS standards things (not just DNSOP)

- DNSOP tinkering
  - stateful-DNS
  - DNS-over-TCP
  - Extended error
- DNS privacy WG
- DNS over HTTPS WG (and then DNS over QUIC?)
- Homenet WG naming architecture
- Etc.

Who wants to standardize these documents and technologies, and why? What good/bad thing happens to the internet if the IETF says no? If the IETF says yes?

….Why worry about “the standard” at all?

Interoperability is what makes it “the internet”

We think this means people are going to extend the DNS protocol whether we like it or not, so they should document that, and having the protocol standards and the assorted commentaries on them as part of one document series and one conversation-- RFCs written and vetted in the IETF-- seems valuable. But we have to recognize that this costs effort, and risks irretrievable fragmentation.

Discuss.