The ISP Column
A monthly column on things Internet

June 2019
Geoff Huston

Network Protocols and Their Use

1 – Case Studies

In June I participated in a workshop, organized by the Internet Architecture Board, on the topic of protocol design and effect, looking at the differences between initial design expectations and deployment realities. These are my impressions of the discussions that took place at this workshop.

In this first part of my report I'll report on the case studies of two protocol efforts and their expectations and deployment experience. These are the Border gateway Protocol (BGP) and the security extensions to the DNS (DNSSEC).

BGP

Routing protocols have been a constant in the Internet, and BGP is one of the oldest still-used protocols. Some aspects of the original design appear to be ill-suited to today's environment, including the general approach of session restart when unexpected events occur, but this is merely a minor quibble. The major outcome of this protocol has been its inherent scalability. BGP is a protocol designed in the late 1980's, using a routing technology described in the mid 1950's, and first deployed when the Internet that it was used to route had less than 500 component networks (Autonomous Systems) and less than 10,000 address prefixes to carry. Today BGP supports a network which is approaching a million prefixes and heading to 100,000 ASNs. There were a number of factors in this longevity, including the choice of a reliable stream transport in TCP, instead of inventing its own message transport scheme, the distance vector's use of hop-by-hop information flow allowing various forms of partial adoption of new capabilities without needing all-of-network flag days and a protocol model which suited the business model of the way that networks interconnected. These days BGP also enjoys a position of entrenched incumbent which itself is a major impediment to change in this area, and the protocol's behavior now determines the business models of network interaction rather than the reverse.

This is despite the obvious weakness in BGP today, including aspects of insecurity and the resultant issue of route hijacks and route leakage, selective instability and the bloating effects of costless advertisement of more specific address prefixes.

Various efforts over the part thirty years of BGP's lifetime to address these issues have been ineffectual. In each of these instances we have entertained design changes to the protocol to mitigate or even eliminate these weaknesses, but the consequent changes to the underlying cost allocation model or the business model or the protocol's performance are such that change is resisted. Even the exhortation for BGP speakers to apply route filters to prevent source address spoofing in outbound packets, known as BCP 38, is now twenty years old, and is ignored by the collection of network operators to much the same extent that is was ignored twenty years ago, despite the massive damage inflicted by a continuous stream of UDP denial of service attacks that leverage source address spoofing.

The efforts to secure the protocol are almost as old as the protocol itself, and all have failed. Adding cryptographic extensions to BGP speakers and the protocol in order to support verifiable attestations that the data contained in BGP protocol packets is in some sense "authentic" rather than synthetic impose a level of additional cost to BGP that network operators appear to be unwilling to bear. The issues of security itself, where it can only add credentials to "good" information, imply that universal adoption is required if we want to assume that everything that is not "good" is necessarily "bad" only adds the formidable barriers of universal adoption and the accompanying requirement of lowest bearable cost, as every BGP speaker must be in a position or accept these additional costs.

We have not seen the end of proposals to improve the properties of BGP, both in the area of security and in areas such as route pruning, update damping, convergence tuning and such. Even without knowledge of the specific protocol mechanisms proposed in each case, it appears they such proposals are doomed to the same fate as their predecessors. In this common routing space cost and benefit are badly aligned, and network operators appear to have little in the way of true incentive to address these issues in the BGP space. The economics of routing is a harsh task master and it exercises complete control over the protocols of routing.

DNSSEC

If BGP is a mixed story of long-term success in scaling with the Internet and at the same time a story of structural inability to fix some major shortcomings in the routing environment it is interesting to compare this outcome with that of DNSSEC.

DNSSEC was intended to address a critical shortcoming to the DNS model, namely through the introduction of a mechanism that would allow a client of the DNS to validate that the response that the DNS resolution system has provided is authentic and current. This applies to both positive and negative response, so that when a positive response is provided, this is verified as a faithful copy of the data that is served by the relevant zone's authoritative name servers, and where a negative response is provided, then the name really does not exist in the zone. We have all heard of the transition of the Internet from an environment of overly credulous mutual trust and lack of skepticism over the authenticity of the data we receive from protocol transactions that occur over the Internet to one of suspicion and disbelief, based largely on the continual abuse of this original mutual trust model. A protocol that would be clearly informative of efforts to identify when the DNS is being altered in various ways by third parties would have an obvious role and would be valued by users. Or so we thought. DNSSEC was a protocol extension to the DNS was intended to provide exactly that level of assurance and it is a complete and utter failure.

In terms of protocol design stories of failure are as informative, or even more so, as stories of success. In the case of DSNSSEC the stories of its failure stretch across its twenty years of progressive refinement.

The initial approach, described in RFC 2535, had an unrealistic level of inter-dependency such that a change in the apex root key required a complete rekeying of all parts of the signed hierarchy. Subsequent efforts were directed to fix this "re-keying" problem. What we have today is more robust, and within the signed hierarchy rekeying can be performed safely, but the root key roll still presents major challenges. Every endpoint in the DNS resolution environment that performs validation needs to synchronize itself with the root key state as its single "trust anchor". This use of a single trust point is both a feature and a burden on the protocol. It eliminates many of the issues we observe in the Web PKI, where multiple trusted CAs create an environment that is only as good as the poorest quality CA, which in turn destroys any incentive for quality in this space. Every certificate is equally trusted in that space. In a rooted hierarchy of trust all trust derives from a single trust entity, which creates a single point of vulnerability and also creates a natural point of monopoly. It is a deliberate outcome that the root key of the DNS is managed by the IANA in a role of trustee representing public interest.

Yet even with this care and attention to a trusted and secure root, DNSSEC is still largely a failure, particularly in the browser space. The number of domains that use DNSSEC to sign their zone are not high, and the uptake rate is not a hopeful one. From the perspective of a zone operator the risks of signing a zone are clearly evident whereas the incremental benefits are far less tangible. From the perspective of the DNS client a similar proposition is also the case. Validation imposes additional costs, both in time to resolve and in the reliability of the response, and the benefits are again less tangible.

Perhaps two additional comments are useful here to illustrate this point. When a major US network operator first switched on DNSSEC in their resolvers the domain name nasa.gov had a key issue and could not be validated. The DNSSEC model is to treat validation failure as ground to withhold the response. So nasa.gov would not be resolved by these resolvers. At the time there was a NASA activity that had generated significant levels of public interest, and the DNS operator was faced with either turning DNSSEC off again or adding the additional measure of manually maintained "white lists" where validation failure would be ignored, adding further costs to this decision to support DNSSEC validation in their resolution environment. The second issue is where validation takes place. So far, the role of validation of DNS responses has been placed on the recursive resolver, not the user. If a resolver has successfully validated a DNS response it sets the AD bit in the response to the stub resolver. Any man-in-middle that sites between the stub resolver and the recursive resolver can manipulate this response if the interaction is using unencrypted UDP for the DNS. If the zone is signed and validation fails then the recursive resolver reports a failure of the server, not a validation failure. In many cases (more than a third of the time) the stub resolver interprets this as signal to re-query using a difference recursive resolver and the critical information of validation failure and the implicit signal of DNS meddling is simply ignored.

Surely there is a market for authenticity in the name space? The commercial success of the WebPKI, which was an alternative approach to DNSSEC, appears to support this proposition. For many years while name registration was a low value transition, the provision of a domain name certificate was a far more expensive proposition, and domain holders paid. The entrance of free certificates into the CA market was not an observation of the decline in value of this mechanism of domain name authentication but an admission of the critical importance of such certificates in the overall security stance of the Internet, and a practical response to the proposition that security should not be a luxury good but be accessible to all.

Protocol Failure or Market Failure?

Why has DNSSEC evidently failed? Was this a protocol failure or a failure of the business model of name resolution? The IETF's engagement with security has been variable to poor, and the failure to take a consistent stance with the architectural issues of security has been a key failure here. But perhaps this is asking too much of the IETF.

The IETF is a standardization body, like many others. Producers of technology bring their efforts to the standards body, composed of peers and stakeholders within the industry, and the outcome is intended to be a specification that serves two purposes. The first is to produce a generic specification that allows competitive producers to make equivalent products, and the second is to produce a generic behavior model that allows others to build products that interact with this standard product in predictable ways. On both cases the outcome is one that supports a competitive marketplace, and the benefit to the consumer is one based on the disciple of competitive markets.

But it is a stretch to add "architecture" to this role, and standards bodies tend to get into difficulties when they attempt to take a discretionary view of the technologies that they standardize according to some abstract architectural vision. Two cases illustrate this issue for the IETF. When Network Address Translators (NATs) appeared in the early 1990's as a means of forestalling address exhaustion the IETF deliberately did not standardize this technology on the basis that is did not sit within the IETF's view of the Internet's architecture. Whatever the merits or otherwise of this position, the outcome was far worse than many had anticipated. NATs are everywhere these days, but they have all kinds of varying behavior because NAT developers had no standard IETF specification of behavior to refer to. The burden has been passed to the application space, because applications that require an understanding of the exact nature of the NAT (or NATS that they are behind) have to also use a set of discovery mechanisms to reveal the nature of the address translation model being used in each individual circumstance. The other case I'll use is that of Client Subnet in the DNS. Despite a lengthy prolog to the standard specification that the IETF did not believe that this was a technology that sat comfortably in the IETF's overall view of a user privacy architecture and should not be deployed, Client Subnet has been widely deployed, and in too many cases has been deployed as a complete client identity. For the IETF a refusal to standardize in architectural ground has its negative consequences if the deployment of the technology occurs in any case, and a reluctant version of standardization despite such architectural concerns again has its negative consequences, in that deployers are not necessarily sensitive to such reluctance in any case.

Even if the IETF is unable to carry through with a consistent architectural model, why is DNSSEC a failure and why has the WebPKI model the incumbent model for web security, despite its obvious shortcomings? One answer to this question is the first adopter advantage. The WebPKI was an ad hoc response by browsers in the mid-1990s to add greater level of confidence in the web. If domain name certificates generated sufficient levels of trust in the DNS (and routing for that matter) that the user could be confident that the site on their screen was the site that they intended to visit, then this was a sufficient and adequate answer.

Why change it? What could DNSSEC use add to this picture?

Not enough to motivate adoption it would seem. In other words, the inertia of the deployed infrastructure leads to a first adopter advantage. An installed base of *a protocol that is good enough for most uses* is often enough to resist adoption of *a better protocol.* And when it's not clearly better but just *a different protocol,* then the resistance to change is even greater.

Another potential answer lies in centralization and cartel behaviors. The journey to get your Certification Authority into the trusted set of the few remaining significant browsers is not easy. The CAB forum can be seen both as a body that attempts to safeguard the end user's interest by stipulating CA behaviors that are an essential set of preconditions to being accepted as a trusted CA and a body that imposes barriers to entry by potential competitive CAs. From this perspective DNSSEC, and DANE, can be views as an existential threat to the CA model and resistance to this threat from the CAB forum is entirely predictable and expected. Any cartel would behave in the same manner.

A third answer lies in the business model of outsourcing. The DNS is often seen as a low maintenance function. A zone publisher has an initial workload of setting up the zone and its authoritative servers, but after that initial setup the function is essentially static. A DNS server needs no continual operational attention to keep it responding to queries. Adding DNSSEC keys changes this model and places a higher operational burden on the operator of the zone. CA's can be seen as a means of outsourcing this operational overhead. It is a useful question to ask why the CA market still exists and why are there still service operators who pay CAs for their service while free CAs exist. Let's Encrypt uses a 90-day certification model, so the degree to which the name security function is effectively outsourced is limited. There is a market for longer term certificates that are a more effective way of outsourcing this function, and the continuing existence a large set of CAs who charge a price points to the continuing viability of this market.

Even though DNSSEC has largely failed in this space so far, should the IETF have avoided the effort and not embarked on DNSSEC in the first place? I would argue against such a proposition.

In attempting to facilitate competition in the Internet's essential infrastructure the IETF is essentially an advocate for competitive entrants. Dominant incumbents have no essential need to conform to open standards, and in many situations, they use their dominant position to deploying services based on technologies that are solely under their control, working to achieve a future position to complement the current situation. Most enterprises who obtain a position that allows the extraction of monopoly rentals from a market will conventionally seek to use the current revenue stream to further secure their future position of monopoly. In the IT sector, when pressed such dominant actors have been known to use crippling Intellectual Property Rights conditions to prevent competitors reverse engineering their products to gain entry to the market. In this light of such behaviors, the IETF acts in ways similar to a venture capital fund, facilitating the entrance of competitive providers of goods and services through open standards. Like any venture capital fund there are risks of failure as much as there are benefits of success, and the failures should not prevent the continual seeking of instances of success.

While I am personally not ready to write DNSSEC off as a complete failure just yet, there is still much the IETF can learn about why it spend many years on this effort. The larger benefits of such activities to the overall health of a diverse and competitive marketplace of goods and services in the Internet is far more important than the success or otherwise of individual protocol standardization efforts.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net