The ISP Column
A monthly column on things Internet

June 2019
Geoff Huston

Network Protocols and Their Use

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.

Lessons from the Past

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.

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.

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.

Deployment Considerations

Do we really understand the expectations of protocols? What do we expect? Are these expectations part of a shared understanding, or are there a variety of unique and potentially clashing expectations? Do we ever look back and ask whether we built what we had thought we were going to build? Did anyone talk to the deployers and operators and competitors to understand their expectations, requirements and needs? In many working groups the loudest voices and the strongest held opinions might dominate a group's conversation, but this is not necessarily reflective of the broader position of interested parties, and not necessarily reflective of the path that represents the greatest common benefit. The strongeszt

supporters of a single domain of interoperable connectivity are often new entrants and incumbents may have an entirely different perspective of the scope and expectations of a standardization effort.

Not only is this a consideration when embarking on standardization of a new protocol or a new tool element, but similar considerations apply to efforts to augment or change a standard protocol. Existing users may oppose the imposition of additional costs to their use of a protocol that appear to unfairly benefit new entrants. Change by its very nature will always find some level of opposition in such forums.

Perhaps one possible IETF action could be to avoid working on refinements and additions to deployed protocols, as this works against the interests of the deployed base and also sends a negative signal about the risks of early adoption of an IETF protocol. On the other hand, the IETF is not working in isolation, and the market itself would resist the adoption of protocol changes if those changes had no substantive bearing on the functionality, integrity or cost of the deployed service. In other words, if the augmentations offer no benefits to the installed base other than opening up the service realm to more competitors it is entirely reasonable to anticipate resistance towards such changes. A direction to the IETF to stop work on protocol refinements may well be a direction to stop working on ultimately futile efforts, and instead spend its available resources working in potentially more productive spaces, as the market will perform such choices between sticking to an existing protocol or adopting change in any case. But many items or work are started in the IETF with confident expectations of success, and "no" is a very difficult concept in an open collaborative environment. It does not need complete agreement, or even a rough consensus of the entire community to embark on activity. The more typical threshold is a cadre of enthusiasm. Whether its individuals or some corporate actors make no substantive difference in such circumstances.

This lack of critical ability to select a particular path of action and make choices between efforts has proved to be a liability at times. The standardization of numerous IPv6 transition mechanisms appeared to make a complex task far harder for many operators. The continuing efforts to tweak the IPv6 protocol appears to act against the interests of early adopters and a sense of delay and caution has become a widespread sentiment among network and service operators.

Scale has been a constant factor in deployment considerations. Protocols that can encompass increasing scope of deployment without imposing onerous costs of early adopters who are forced to keep up with the growth pressures being imposed by later entrants tend to fare better than those that impose growth costs on all. The explosive growth of Usenet news imposed escalating loads on all providers, and ultimately many dropped out. The broader issue of the scalability and cost of information flooding architectures cannot be ignored as an important lesson from this particular example.

Many protocols require adjustment to cope with growth. A good example here is the size of the Autonomous Number field in BGP. The original 16-bit field was running out and it was necessary to alter BGP to increase the size of this field. One option is a "flag day" where all BGP speakers shift to use a new version of the protocol. Given the scope of the Internet this has not been a realistic proposition for many years and probably many decades. The alternative is piecemeal adoption, where individual BGP speakers can choose to deploy a 32-bit ASN capable version and interoperate seamlessly with older BGP speakers. In general, where change is necessary for a deployed protocol, piecemeal deployments that are backward compatible with the existing user base will have far better prospects than those which are less flexible. In the early days of designing what was to become the IPv6 protocol there were various wish lists drawn up. "Backward compatibility" was certainly desired in this case, but no robust way of achieving this was found, and the protracted transition we are experiencing uses a somewhat different approach of coexistence, in the form of the dual stack Internet. Coexistence implies that any network cannot rid itself of a residual need for IPv4 services while any other network is still only operating an IPv4-only network. The entire transition process stalls on universal adoption, where the late adopters appear to claim some perverse form of advantage in the market through deferred cost of transition.

Is the IETF's conception of "need" and "requirement" distanced from the perspectives of operators and users? Should the IETF care when operators or users don't? Transport Layer Security (TLS) is a good illustration here. While the network was largely a wired network it was evident that users trusted network

operators with their traffic, and efforts to encrypt traffic did not gain mainstream appeal. TLS only gained gained traction with the general adoption of WiFi, as the idea of eavesdropping on radio was easy to understand. And at this point the message of the need for end-to-end encryption had a more receptive audience. Should the IETF have waited until the need was obvious, or were its early actions useful in having a standard technology specification already available when user demand was exposed? It is hard to believe that the IETF has superior knowledge of the requirements of a market than those actors who either service that market or intend to invest in servicing that market. Having the IETF wait until it makes a clear judgement as to need runs the risk of only working on already deployed technology. At this point the value proposition of an open and interoperable standard is one that exists for all but the original developers and early adopters.

How do standards affect deployment? HTTPS is an end to end protocol that can be used to drive through various forms of firewalls and proxies. Packages that embed various services into of HTTPS sessions, including IP itself have existed for years, although the lack of applicable standards have meant that their use was limited to those who were willing and able to install custom applications on their platforms. The recent publication of RFC 8484 that described the technique of DNS over HTTPS (DOH) was more a case of formalizing an already well understood tunneling concept than representing some new invention. The existence of an IETF standard document effectively propelled this technology into a form of legitimacy, transforming it from just another tool in the hackers toolbox into something that some mainstream browser vendors are intending to fold into their product. The standard in this case is seen as a precursor to widespread adoption. That should not imply that there is broad agreement about the appropriateness of the standardization or broad agreement with the prospect of broad deployment. DOH has been a story of emerging difference of expectations. Some browser vendors appear to be enthusiastic about DOH as an enabler of faster service with greater control placed into the browser itself, lifting the name resolution function out from the platform and placing it into the application. However, the DNS community is not so clearly on board with this, seeing DOH as a potential threat to the independent integrity of the DNS as a distinct element of a common and consistent Internet infrastructure. Once the name resolution function is pushed deeply into the application what's to prevent applications from customizing their view of the name space? An important value of a single communications network resides within the concept of a single referential framework, where my reference to some network resource can be passed to you and still refer to the same resource. Should the IETF not work on technology standards that head down paths that could potentially lead to undermining the cohesive integrity of the common Internet namespace? Or are such deployment consequences well outside the responsibility of the IETF?

Deployment of technologies has exposed many tussles in the Internet. One of the major issues today is the tussle between applications and platforms. Today's browsers are now a significant locus of control, exercising independent decisions over transport, security, latency, and the name space, which collectively represent independent control over the entire user experience. Why should the IETF have an opinion one way or the other on such matters? If you take the view that a role of standards as to facilitate open competition between providers, then the issue in this space lies in the inexorable diminution of competition in the Internet. It appears that if one can realize unique economies of scale, and greater scale generates greater economies then the inevitable outcome is concentration in these markets. One of the essential roles of the IETF is diminished through this concentration within the deployment space, and the IETF runs the risk of being relegated to rubber-stamping technologies that have been developed by incumbents.

How can the IETF we measure the level of concentration in a market? If the IETF were to claim that they had an important role in supporting competition in decentralized markets, then how exactly would the IETF execute on this objective? What would it need to do? Is protocol design and standardization relevant or irrelevant to the industry composition of deployment that breeds centralization? Can the IETF ever design a protocol that would be impossible to leverage in a centralized manner? This resistance to concentration within the Internet appears to be an unlikely mission for the IETF. The Internet's business models leverage inputs and environments to create advantage to incumbent at-scale

operators. It would be comforting to think that the protocols used, and their properties are largely orthogonal to this issue.

However, there is somewhat more at play. Standardization occurs during the formative stages of a technology, and this may be associated with deployment conditions that include early adopter advantages. If such advantages exist, then the rewards to such early adopters May be disproportionately large. This engenders positive market sentiment which motivates the early adopter to defend its unique position and discourage competition. Early adopters head to the IETF to shape emerging protocols and influence their intended entrance into the market. Their interests in the standardization process is not necessarily to generate a technology specification that facilitates opening up the technology to all forms of competitive use. Often their interests lie in the production of complex monolithic specifications replete with subtle interdependences and detail. Trying to position the IETF work to encourage competition by producing simple specifications of component elements that are readily accessible runs counter to the interests of early adopters and subsequent incumbents.

There is an entire world of economic thought on market dominance and competition, and it becomes relevant to this consideration about protocols and centrality in the Internet. Is *big* necessarily *bad*? Is *centralization* necessarily *bad*? Or is the current environment missing some key components that would've controlled and regulated the dominant incumbents? In many ways it seems that we are re-living the Gilded Age of more than a century past. There is a feeling of common unease that the Internet, once seen as a force for good in our society, has been entirely captured by a small clique who are behaving in manner consistent with a global cabal. The response to such feelings of unease over the ruthless exploitation of personal profiles in the deployment space is to seek tools or levers that might reverse this situation. The tools may include law and regulation, the passage of time, new protocols, educating users, or new vectors of competition. In many ways this common search for a regulatory lever is largely ineffectual, as the most effective response to market dominance often is sourced from the dominant incumbent itself.

Security and Privacy

These days any form of consideration about the Internet and its technology base needs to either address the topic of security and privacy in all its forms, or explicitly explain this glaring omission. Obviously, this workshop headed directly into this space, asking whether the IETF was looking at topical and current threat models, and also asking about likely evolution in this space.

Exhortations about security practices for service operators made through standards bodies are often ineffectual in isolation. RFC 2827 is almost 20 years old, and it ignored by network operators to about the same extent that it was ignored at the time of its publication. It may be better known as BCP 38, or packet filtering to prevent source address spoofing in IP packets. It's important because there is a class of DDOS attack using UDP amplification where the UDP response is far larger than the query. It's a fine practice and we should all be doing this form of filtering. Twenty years later the attacks persist because the filtering is just not happening. What may make such forms of advice more effective is the association of some form of liability for service operators, or explicit obligations as part of liability insurance. In isolation, advice relating to security measures is often seen as imposing cost without immediate direct benefit, and in circumstances such as this case, where the defensive approach is only effective when most operators undertake the practice, benefits for early adopters are simply not present.

Another example is the standardization of Client Subnet extensions in DNS. Despite the standard specification RFC 7871 containing the advice that this feature should be turned off by default and users be permitted to opt out, this has not happened. This is in spite of the potential for serious privacy leak through attribution of DNS queries to end users.

The environment of attacks escalates, as the growing population of devices allows the formation of larger pools of co-opted devices that in concert can mount massive DDOS attacks. Given our inability to prevent such attacks from recurring, the reaction has been the formation of a market in robust content

hosting. As the attacks increase in intensity the content hosting operators require larger defensive measures and economies of scale in content hosting come into play. The content hosting and associated distribution network sector is increasingly concentrated into a handful of providers. In many ways this is a classic case of markets identifying and filling a need. The distortion of that market into a very small handful of providers is a case of economies of scale coming into play. As with the CA market, the market has now seen the entrance of zero cost actors, which has significantly lifted the barrier to further new entrants in this market. What remains now appears to be simply a process of further consolidation in the market for content hosting.

The threat model is also evolving. RFC3552, published in 2003, explicitly assumed that the end systems that are communicating have not themselves been compromised in any way. Is this a reasonable assumption these days? Can an application assume that the platform is entirely passive and trustable, or should the application assume that the underlying platform may divulge or alter the application's information in unanticipated ways. To what extent can or should applications lift common network functionality into the user space and deliberately withhold almost all aspects of a communication transaction from other concurrently running applications, from the common platform and from the network? Do approaches like DOH and QUIC represent reasonable templates for responding to this evolved threat model? Can we build protocols that explicit limit information disclosure when one of the ends of the communication may have been compromised?

Is protocol extensibility a vector for abuse and leakage, such as the Client Subnet DNS extension in the DNS, or the session ticket in TLS?

And where are our points of trust to allow us to validate what we receive? As already pointed out DNSSEC is not faring well, and the major trust point is the WebPKI. Unfortunately, this system suffers from a multiplicity of indistinguishable trust and our efforts to detect compromise have shrunk to logging, in the form of Certificate Transparency. Such a measure is not responsive in real time and rapid attacks are still way too effective.

A single trust anchor breeds a natural monopoly at the apex and across the diversity of the global Internet there is a lot of distrust in that single point, particularly when geopolitics enters the conversation. This single trust broker is a natural choke point and is one that tends to drift towards rent seeking if operated by the private sector and distrust if operated by the public sector. Designs for trust need to take such factors into account.

The issue of security popups in the browser world vs silent discard of the response in the DNSSEC world offers two views of security management. Placing the user into the security model leads to lack of relevant information and an observed tendency for the user to accept obviously fraudulent certificates because of no better information. From that perspective removing the user from the picture improves the efficacy of the security measure. On the other hand, there is some disquiet about the concept of removing the user from security controls. Giving the user no information and no ability to recognize potentially misrepresenting situations that may them seems to be a disservice to the user.

Do our standards promote and encode the "state of the art" as a means of shedding liability for negligence while still acknowledging that the state of the art is not infallible? Or do they purport to represent a basic tenet of security that is correctly executed is infallible?

The Internet of (insecure) Things is an interesting failure case, and the predatory view of the consumer often distracts from the ethos of care of the customer and the safeguarding customer's enduring interests. Grappling with conformance to demanding operational standards in a low cost highly diverse and high-volume industry is challenging. Perhaps more so is the tendency of the IETF to develop many responses simultaneously and confront the industry with not one but many measures. Already we've seen proposals that use some level of manufacture cooperation, such as nesting public/private key pair, QR code, MUD profile or boot server handshake.

A safe mode of operation would require that the device cannot cold start, nor even continue to operate without some level of handshake. Is this realistic? Will manufacturers cooperate? Will this improve the overall security of the IoT space. Are these expectations of manufacturers realistic? Will a kickstart IoT toothbrush comply with all these requirements? Will these requirements impose factory costs that make the device prone to manufacturer errors and increase the costs to the consumer without any change in the perceived function and benefit of the device? An IoT toothbrush will still brush teeth irrespective of the level of conformance to some generic standard security profile. The failure in the October 2016 botnet DNS attack that used readily compromisable webcams was not a failure of information or protocol. It was a failure of markets, as there was no disincentive to bring to market an invisibly flawed, but cheap, and otherwise perfectly functional product. We tend to see the IoT marketplace as a device market. In contrast, effective device security is an ongoing relationship between the consumer and the device manufacturer and requires a service model rather than a single sale transaction.

The prospect of regulatory impost to provide channels to the retail market that include conformance to national profiles is nigh on certain. Will the inevitable diversity of such regional, national or even state profiles add or detract from the resultant picture of IoT security? Will we end up with a new marketplace for compliance that offers an insubstantial veneer of effective security for such devices? It's very hard to maintain a sunny optimistic outlook in this space.

Human behavior also works against such efforts as well. Our experience points to an observation that users of a technology care a whole lot less about authentication and validation than we had assumed was the case. Most folk don't turn on validation of mail, validation of DNS responses and similar, even when they had access to the tools to do so. When we observed the low authentication rates post-deployment, our subsequent efforts to convince users to adopt more secure practices were ineffectual. Posters in the Paris Metro informing metro users as to what makes a password harder to guess really have not made an impact. In the consumer market users don't understand security and don't value such an intangible attribute as part of a product or service.

Safegarding privacy is a similarly complex space. The last decade has seen the rise in surveillance capitalism, where the assembling of individual profiles consumers has become the cornerstone of many aspects of today's Internet. Many products and services are provided on the Internet free of charge to the user. The motivation to provide such free services comes from the reverse side of this market, where the tool of service is used to assist in the generation of a profile of the individual user, which is then sold to advertisers. Our digital footprint can provide a rich vein of data to fuel this world of surveillance capitalism. Whether its our browser history, logs of DNS queries, our mailboxes, search history, or documents, e-book purchases and reading patterns, all of this data can be converted into information that has monetary value. Our attitude to this activity is not exactly consistent. On the one hand we appear to be enthusiastic consumers of free-of-charge products and services and all too willing to dismiss reports of data mining on the basis that individually none of us have anything to hide. At the same time, we all have experienced those disconcerting incidents where the delivered ad mirrors some recent browsing topic or received email. Why is privacy a common concern when our actions appear to indicate that we are willing to trade it for the provision of goods and services?

One reason for this concern is when a principle of informed consent is violated. Protocols, products and services should not facilitate unintended eavesdropping on a user's actions and activities. When they leak personal information without such informed consent there is a reasonable reaction over what is perceived to be unacceptable surveillance. Another reason lies in the inherently asymmetric nature of the market of personal profile data. Individual users tend to undervalue their profile data, and the relationship with the consumers of such data tends to be exploitative of individual users.

The IETF's position on privacy has strengthened since the publication of RFC 7258 in May 2014, and the IETF's expectation is to go to some lengths with information management in its protocols to contain what is now seen as gratuitous leakage. This includes measures such as query name minimization in DNS queries and encryption of the SNI field in TLS handshakes.

It would be good to think that we have finally stopped using the old security threat model of the malicious actor in the middle. We have more complex models that describe secure and/or trusted enclaves, which an unknown model of the surrounding environment. Is this device security or really a case of "data security"? We need to associate semantics with that data and describe its access policies to safeguard elements of personal privacy.

Where now?

The internet faces many challenges these days, and while many of these challenges are the consequence of the Internet's initial wild and rapid success, few of these challenges have the same intrinsically optimistic tenor as compared to the challenges of the earlier Internet. We see an increasing capable and sophisticated set of threats coming from well-resourced adversaries. The increasing adopting of Internet-based services in all parts of our world increases the severity of these threats. We also see increasing consolidation by a shrinking set of very large global enterprises. Social media, search, cloud services and content are all offered by a handful of service operators and effective competition in this space is not merely an illusory veneer but has disappeared completely. The increasing dominance of many parts of the Internet by a small set of entrenched incumbents raises the obvious questions about centrality of control and influence, as well as the very real questions about the true nature of competitive pressure in markets that are already badly distorted.

For the IETF this poses some tough questions. Is the IETF there only to standardize those technology elements that these entrenched incumbents choose to pass over to an open standardization process to simply improve the economies and efficiency of their lines of supply while excluding some of their more important technology assets? If the IETF feels that this situation of increasing concentration and the formation of effective monopolies in many of these activity areas calls for some remedial action, then is it within the IETF's areas of capability or even within its chosen role to do anything here?

Some ten years ago the IAB published RFC 5218, on "What Makes a Successful Protocol". Much, if not all, of that document still holds today. The basic success factor for a protocol is for it to meet a real need. Other success factors include incremental deployment capability, open code, open specification and unrestricted access. Successful protocols have few impediments to adoption and address some previously unmet need. RFC 5218 also used a category called *wild success*:

> "… a "successful" protocol is one that is used for its original purpose and at the originally intended scale. A "wildly successful" protocol far exceeds its original goals, in terms of purpose (being used in scenarios far beyond the initial design), in terms of scale (being deployed on a scale much greater than originally envisaged), or both. That is, it has overgrown its bounds and has ventured out "into the wild"." [RFC 5218]

One view is that for the IETF, success and wild success are both eminently desirable. The environment of technology standardization has elements of competitive pressure, and standards bodies want to provide an effective platform for protocol standardization that encourages both submissions of work to be considered by the standardization process and through its standards imprimatur is able to label a technology a useful and useable. For the IETF to be useful at all it needs to be able to engender further wild success in the protocols it standardizes. So there is a certain tension in the propositions that the IETF should pursue a path that attempts to facilitate open and robust competiution and eschew standardizing protocols that lead to further concentration in the market and the position that in order to maintain its value and relevant the IETF should seek to associate itself with successful protocol, irrespective of the market outcomes that may result.

Some of the tentative outcomes of this workshop for me have been:

- Technologies get deployed in surprising ways, which can have unintended consequences in threat models, surveillance capability and user privacy
- The focal point of technology and service evolution is moving up the stack, and applications are now taking responsibility for their own services, transport, security, naming context and similar.
- Perceived needs drive deployment, not virtue!

- Interoperability continues to be important but what are the interfaces that require standardization?
- With the Internet now the mainstream of communications, the support ecosystem is populated with more diverse actors and interests. IETF commentary could be helpful at this point, but by whom and to whom?
- Specific subject issues, such as DDOS, IOT, Spam, DNS, regulation, and centralization, are the topic of many challenging conversations, but none of these issues have easy resolution, and none are resolvable solely within the purview of the IETF.

What should the IETF do?

It is highly likely that the IETF will adopt a highly conservative position to such challenging questions and simply stick to doing what the IETF does best, namely, to standardize technologies within its areas of competence, and let others act as they see fit. The IETF does not define the Internet, nor is it responsible for either the current set of issues or the means of their solution, assuming that solutions might exist. The IETF is in no position to orchestrate any particular action across such a diversity and multiplicity of other actors here, and it would probably be folly for the IETF to dream otherwise.

No doubt the IETF will continue to act in a way that it sees as consistent with the interests of the user community of the Internet. No doubt it will continue to work on standardizing protocols and tools that proponents in the IETF believe will improve the user experience and at the same time attempt to safeguard personal privacy. It is difficult to see circumstances where the IETF would act in ways that are not consistent with such broad principles.

At the same time, I don't see the IETF claiming responsibility for the negative consequences of the use of protocols that it has standardized. While this may seem to be the IETF selectively absolving itself from some form of blame, it is a pragmatic position for the IETF to take. Although the IETF standardized many of the technologies in use in today's Internet, it cannot be held directly responsible for the way in which such technologies are used and the negative consequences of such use. Ultimately it is up to those who deploy products and services on the Internet to be responsible to their consumers and the broader societal and physical environment in which they operate.

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