August 2022
Geoff Huston

Notes from IETF 114

IETF 114 was held in the last week of July 2022 as a hybrid meeting, with the physical meeting being held in Philadelphia. Here's my notes on topics that attracted my interest from the week.

IEPG

The IEPG meetings are held each Sunday prior to the IETF week. These days these IEPG meetings have what could at best be described as an eclectic mix of material, generally on topics relating to current areas of focus in the IETF ads they relate to operational matters.

ROV

The first of these items was a presentation by Koen van Hove on Route Origination Validation measurements. In this case the measurement testbed used a valid aggregate route advertisement and a more specific route advertisement that was invalid within the parameters of the RPKI framework. The aggregate route directs traffic to a VM in Amsterdam, while the more specific, which was invalid within the framework of the RPKI directed traffic to a VM in Sydney.

It's an interesting experiment, but one that I feel has a couple of very significant weaknesses. Firstly, it really needs to be augmented with a large-scale testing regime. The presentation looked at some 2,000 tests without specifying the locations of the test points. Secondly this test regime is not all that helpful in understanding which networks perform RPKI invalid dropping, and the use of mix of aggregate and more specific routes add much in the way of complexity to understanding what is going on. For example, imagine a multi-network path that traverses a set of ROV-Invalid dropping and non-dropping along the path. The path may follow the aggregate then the more specific, then the aggregate route and so on as it traverses through the inter-AS space. BGP was not designed to cope cleanly with such information propagation models and the results are challenging to interpret. (I must admit to some personal bias, here as at APNIC labs we devised a similar measurement regime that is intended to identify the networks that measure the level of ROV deployment and the networks that perform ROV-Invalid dropping (https://stats.labs.apnic.net/rpki). In the APNIC case we used a single route advertisement and regularly flipped the origin validation outcome over time between valid and invalid states. I feel that this scales in a far better way, although AS path issues still tend to blur the outcomes of this measurement.)

DANE Portal

I am still surprised that DANE has had such a lukewarm reception in the Internet. The ability to associate information (such as a public key) with a domain name in a secure and testable manner would seem to me to be a far superior method than the rather haphazard (and often abused) approaches used by existing Certificate Authorities in the Web PKI, and to my mind DANE should've been a more significant success story than has been the case. But reality has gone down a different path. One possible reason for this poor reception of DANE is that it is just all too hard, and a group at George Mason University has devised a tool framework, DANE Portal, intended to make the DANE provisioning process easier.

The basic approach is for this system to manage the process for the user. The DNS zone containing the DANE record is delegated to a DANE Portal server and managed remotely. This form of outsourcing security comes with benefits and risks, in that the household tasks of good key management are now

outsourced to this service provider, but at the same time part of your identity is now controlled by a third party and were that third party to be compromised in any way there are consequences that extend back to the customer. In other words, I'm not sure that "making it easier to do" is the same as "I'll do it for you".

On the other hand, the entire X.509 public key certificate framework is proving to be an inadequate match for the Internet and some decades of adding bandages and splints to a basically ill-suited framework appears to have done little other than expose yet more vulnerabilities. DANE is far less ambitious in its objective, in that it securely associates a digital artefact with a domain name, to the extent that a DNSSEC-verified DANE record is a precise copy of the information that was entered into the zone in the first place. To the extent that these tools provide a way for third parties to validate that claims to be within the space defined by a domain name can be signed and validated via DANE and DNSSEC, then this is a good stop forward in a search for a credential framework that is better suited to the requirements of Internet applications.

IPv6 DST EH

The saga of Extension Headers in IPv6 continues. These extensions to the basic IPv6 packet header have their antecedents in the IPv4 options. These options, including loose source routing, record route, MTU and about 25 others have largely fallen into disuse, and are essentially unmourned in their passing! Some of these options represented a security risk, others were a potential drain on router resources and it turn out that others performed no useful function. However, this did not stop the IPv6 design from including a similar collection of IP-level packet options, packaged in the IPv6 packet header between the common IPv6 header and the transport section. These extension headers include a set of options that need to be examined by all IP-level devices on the packet's end-to-end path (Hop-by-Hop), options that need to be examined only by the destination of the packet (the remote "end") (Destination Option), Fragmentation control, routing directives, security, and specialised support for the HIP and SHIM6 protocols. There was little in the way of testing if these options worked within the Internet for many years, and it was a surprise for many to read RFC 7872 in 2016 to learn that there were significant levels of transmission failure of IPv6 packets with various forms of IPv6 Extension Headers.

This work has promoted other researchers to investigate this behaviour using different test scenarios and the results are, at the very least, confusing. While some experiments report packet drop at levels that are close to comprehensive, others are reporting no appreciable drop at all, such as in this IEPG presentation. This experiment used a non-standard Destination Option intended for use in performance monitoring, and they undertook a set of data transfers between test platforms where the data packets used this extension header.

It seems, in the first instance, to be curious that different ways of observing and measuring the same basic behaviour yields such different outcomes. In looking at the fate of a packet there are a number of causes of drop. A network element may drop the packet in flight (which appears to be more likely for Hop-By-Hop headers than, say for Destination or Fragmentation options). Or the intended destination may drop the packet as it is contrary to local acceptance policies (as may be the case for Destination options). Commonly deployed edge devices (iOS and Android devices) may behave differently to hosted Virtual Machines in this respect.

Irrespective of the precise measured outcomes my takeaway is that it's likely that there are some issues here for IPv6 Extension Headers and the best advice is that they can be used in certain cases and certain environments, but if you are looking for advice with respect to general applicability in all forms of IPv6 environments, then you should check before use, and if you can't check, then avoid them completely.

QUIC

This protocol is both a newcomer to the Internet and an already middle-aged veteran protocol. It was first introduced to the IETF back in 2012 by Google, and Google offered to work with the IETF to standardise this approach. This has subsequently occurred and the QUIC specification has been published as RFC9000 in May 2021. For some years now the Chrome Browser has been reacting to

service directives in the HTTP content headers to inform the client if the server of this content is capable of supporting QUIC. More recently Apple has added support to lookup the DNS using a HTTPS query type, and if there is a directive to use the H3 application-level protocol then the Safari browser will attempt to perform the URL fetch using QUIC (although it is not as simple as that – it appears that this QUIC fetch is limited in safari and only a quarter or so of cases where this is a HTTPS record and it is accessed by the browser result in a QUIC fetch).

APNIC has recently started reporting on a structured measurement of QUIC use. The measurement technique uses a two-step fetch, and this allows the outcomes of QUIC use for the first fetch to be compared with the outcomes of the second fetch. Apple's Safari uses the DNS HTTPS query and will detect and use QUIC on the first fetch. Chrome uses the service directive embedded in the content, so it should detect the server's willingness to use QUIC on the first fetch and switch to use QUIC on the second fetch.

This is indeed what we see, but more puzzling is the observation that the conversion rate to QUIC is very low in both cases. Less than 1 on 4 who are seen to ask for the HTTPS record follow up with a QUIC fetch. In the case of Chrome, it seems that the conversion rate is around 1 in 15 or so for the second fetch. The reason for these low measurements is, as yet, not entirely understood, but it appears to be a combination of browser behaviour and server configuration (if the server is using persistent TLS connections then the second fetch may make use of the open connection rather than start a new QUIC connection).

Slides for the IEPG meeting are at http://www.iepg.org/2022-07-24-ietf114/index.html

Transport

TSV - Transport Area

It seems somewhat odd, but there has been little in the way of standardisation of TCP's various congestion algorithms. It's likely that the Transport Area will chartering a new IETF working group to update the administrative framework for congestion control standardisation, and potentially adopt any proposals that are sufficiently mature for the standards track.

This all sounds like a good idea, but the evidence is that it's hard to get transport specifications through the IETF. It's not just devising a robust and clear specification that will result in multiple interoperable implementations, but to devise a congestion control algorithm that not only provides acceptable performance outcomes, but also devise one that is not overly hostile to other congestion control algorithms. And this objective creates a somewhat more challenging task, namely to devise some standard forms of test environments that can provide a way to evaluate the interoperation of one such congestion control protocol with the existing environment of deployed and in-use congestion control protocols. There is a distinction between documenting what is used as a simple act of recording the current reality and adding the IETF imprimatur of quality to a specification by documenting it as a published RFC.

Some new congestion control protocols have been discussed in detail in the Congestion Control Research Group but the experience to date is that deployment has proceeded without an imposed precondition as a published specification. Is the intent here to attempt constrain the set of congestion control behaviours that are used in the internet? In which case the effort is likely to be ineffectual. Is the intent to enable a non-hostile interaction between set of congestion control behaviours that are in used simultaneously, which gets to a somewhat nebulous proposition of defining more precisely that that term means and how it might be measured and in what context(s).

As in usual in IETF Working Group meetings, there was much enthusiasm in the room for someone to do something without much of an idea of who was to do it or what was to be done!

TCPM - TCP Maintenance and Minor Extensions

The issue of writing up TCP congestion control protocols was illustrated in the subsequent TCPM working group in the work on the specification of the revised CUBIC specification as a proposed standard. Now this is cheating a bit as it is already published as RFC8312, and the working group discussion illustrated the underlying issue of the objective of the standard specification. The current specification in RFC 8312 states that the window multiplicative decrease function is to be 0.7, which is a smaller decrease than the Reno-like behaviour that use a slightly larger window decrease factor of 0.5. The issue here is that there is no clear "right" answer. In some scenarios it makes sense for the TCP session to maintain some pressure on the buffer under contention to avoid being shut out by concurrent TCP sessions, while in other contexts it makes sense to back off further in the hope that this will be more helpful in clearing the congestion condition.

Another example is the work on proportional rate reduction in TCP. RFC6937 was published in 2012 as an experimental congestion control algorithm to be applied during fast recovery. Since then, it is now the default in Linux, FreeBSD, and Netflix-BSD/Rack. Clearly in practical terms it's no longer experimental, so the TCPM working group is revising this specification and publishing it as a proposed standard RFC. Without delving deep into the issues, the question for the working group is whether to stick with a specification that one hopes matches the deployed protocol, or whether the Working Group follows a path to make further tweaks along the way. Obviously the "make further tweaks" approach has won the day!

TCP is a feedback-controlled flow protocol. The feedback to the sender lets it know if its current sending rate is too fast, too slow or balanced to the prevailing conditions. There appear to be two different schools of thought about the operation of this feedback loop. One, typified by BBR, is to react to this feedback signal in a timescale of multiple Round Trip Time intervals (In BBR's case it is 8 RTT intervals). The other, typified by Reno and CUBIC, is to react each RTT interval, making smaller, but continuous adjustments in the sending rate in response to this feedback signal. Reno and CUBIC were both loss-reactive protocols, where the implicit signals relating to the onset of network queuing did not cause a response from the sender. With the introduction of a coupling of the network path queuing state and TCP through the use of Explicit congestion notification (ECN) the option of increasing the sensitivity of the feedback signal was enabled.

The accurate ECN draft (draft-ietf-tcpm-accurate-ecn-20) (which now has a venerable status after seven years as a draft) advocates an even richer feedback signal, changing the ECN signal handling at the receiver. As the draft describes: "ECN was originally specified for TCP in such a way that only one feedback signal can be transmitted per Round-Trip Time (RTT). Recent new TCP mechanisms like Congestion Exposure (ConEx), Data Center TCP (DCTCP) or Low Latency Low Loss Scalable Throughput (L4S) need more accurate ECN feedback information whenever more than one marking is received in one RTT. This document updates the original ECN specification to specify a scheme to provide more than one feedback signal per RTT in the TCP header."

In many ways this is a replay of an earlier conversation about the ACK rate in TCP where the results of using a sparse ACK rate can be contrasted to the approach of ACK every received packet. One issue appears to me be to what extent does a finer level of granularity in the feedback signal add distracting noise? If the problem here is the coupling of the TCP sending rate to the change in the network path queue state, then why not go all the way to BBR and make the flow control reliant on the onset of queuing, rather than on the slightly more nebulous condition of the onset of congestion (whatever that is!).

Routing

Source Address Validation

It's been more than 20 years since source address filtering was described in RFC2827 (May 2000). A common form of DDOS attack is to generate traffic with a forged source IP address and then direct that traffic to a UDP-based server, such as a DNS service, or NTP. The responses will be directed to the

intended victim, and if the responses are larger than the service requests then the server is coopted into the attack as an unwitting amplifier. Source address filtering is simple for single-attached stub networks where a filter is applied at the point of attachment of this stub network to limit all outbound packets to use source addresses drawn from the address set announced by this stub network. Such a scenario is readily automated by combining the packet filter to the routing state.

Can this technique be used in other scenarios? Can an automated process of source address filtering be applied at an arbitrary point of network interconnection within the Internet? This is a far harder question, and the technique of analysing the routing state to collect a collection of reachable addresses that can be used to derive a source address filter set creates overly restrictive outcomes. At the heart of the problem is the issue of asymmetry in packet forwarding in the internet. Routing generates forwarding paths through the internet based on each destination address, and the selected path from A to B may be different from the path from B to A. This implies that the routing system cannot necessarily inform a filter of all the source addresses that may be used by packets passing from A to B. RFC3704 (March 2004) analysed this situation and consider some cases of multi-homed stub networks, but the potential approaches in that document have some aspects that resemble optimistic handwaving! The problem was re-visited in RFC8704 (February 2020) with an "Enhanced" Feasible-Path Unicast Reverse Path Forwarding. This approach is based on attempting to amass the set of all prefixes that could be sourced from an AS (the "customer cone" of an AS) and using this entire set of prefixes as a source address filter for that AS on all interfaces where that AS is referenced in an AS Path on a received route.

The material presented at the SAVNET working group in this IETF represent a couple of difference approaches.

One is in the early steps of a structured approach to the issue, using gap analysis, deriving a problem statement and requirements, and then formulating potential approaches. At this stage the potential approaches contain significant complexity and rely once more on mutual trust between networks to generate customer cones. However, as this process is in its early stages, it's hard to tell where this process may end up and just how effective the outcomes may be.

Another approach leverages the work on routing security and postulates that the collection of ROAs that refer to an AS represent the complete set of prefixes that a network may announce. So if you use the inter-AS relationships that are presented in RPKI ASPA objects, and the originated prefixes as represented in ROAs, then one could construct feasible customer cones for an AS and derive prefix filters from that data. Here the problem is that the ROAS are not generated by the AS, but by the prefix holder and may not correlate to the set of prefixes originated by an AS, It's also the case that both ROAs and ASPA objects are not complete, so any permissive filter constructed in this way would discard traffic that would otherwise be legitimate.

Perhaps it's useful to ask a basic question. Is the objective of source address validation a realistic expectation? Can you really expect the network to detect packets where the source address is not "viable" in the sense that the local network would direct a packet back to that address via an entirely different path that has no direct or even indirect correlation with the ingress path of the received packet? Both approaches have both complexity and some major assumptions, and it's challenging to see where progress in this space might lie.

DNS

DPRIVE

Much of the work so far in adding privacy to the DNS protocols has been concentrated on the path between the edge (the "stub" resolver) and the recursive resolver. In this space we've seen the specification of DNS over TLS (DoT), DNS over HTTPS (DoH) and DNS over QUIC (DoQ). DPRIVE has moved on in the overall agenda to examine the path between recursive resolvers and authoritative nameservers.

It's reasonable to ask "why?" While the path between the stub resolver and the recursive may identify both the IP address of the end client and the name that they are seeking to resolve, the query between the recursive resolver and the authoritative server has no such sensitive information (well that's not quite correct and if EDNS Client Subnet (ECS, RFC 7871) is in use then the client subnet is attached to this query, but ECS is its own problem and to propose encrypted recursive-to-authoritative channels as a response to ECS seems to me to be case of a very expensive solution to what is an avoidable problem! The best I can come up with is the issue of cache poisoning. As long as the vast majority of DNS names are not DNSSEC signed we are vulnerable to various forms of cache poisoning attacks. While query field randomisation has a role to play in making these attacks harder the increasing capacity of the network to bring a large query intensity to bear on a recursive resolver makes the attacks plausible. Encryption shuts down this possibility. Yes, it may be an expensive solution to an uncommon problem, but without any other practical response to this attack vector the uncommon attack may become far more common. No, I'm not convinced either, but I'm hard pressed to come up with anything more plausible!

The specification itself, draft-ietf-dprive-unilateral-probing can be readily summarised as "don't be silly!" If a resolver attempts to open an encrypted session with an authoritative server, then the resolver should remember the outcome, successful or otherwise, for a while. Yes timers, yes, more specific details, but not much else that is vital to interoperability.

The document proposes opportunistic encryption where the server and resolver client do not authenticate each other. This draft proposes IP address level association of server to capability, rather a name-based association of the domain name of the server. The implication is that a server will support encrypted sessions for all the domains it serves. The alternative is to use a name-based scheme where the SNI field of the TLS handshake determines the scope of the queriers covered in the encrypted session. The problem with this name-based approach is that servers that serve a large number of domains would presumably need to support many distinct TLS sessions with each recursive resolver, further adding to the overheads of this approach without much in the way of incremental benefit.

There is an overhead for TLS. It takes additional signalling to set up a session, additional processing to manage the crypto and additional memory to maintain the session states. In a study using a root server and the Google resolver, presented at the DNS OARC meeting after the IETF week (https://indico.dns-oarc.net/event/43/contributions/937/attachments/900/1641/google-tls-abbrev.pdf), it was reported that when the recursive resolver uses ADoT, the received packet count rose by a factor of 2.12 and the transmitted packet count by a factor of 1.54. The bandwidth requirements rose in a similar way, by 1.9 on received traffic and 1.6 on sent traffic. The processing requirements rose by a factor of 1.6. This is quite an overhead for the DNS to absorb and the case to take the DNS in this direction is not exactly clear. It's not a technical consideration, but a cost and benefit issue.

DNSOP

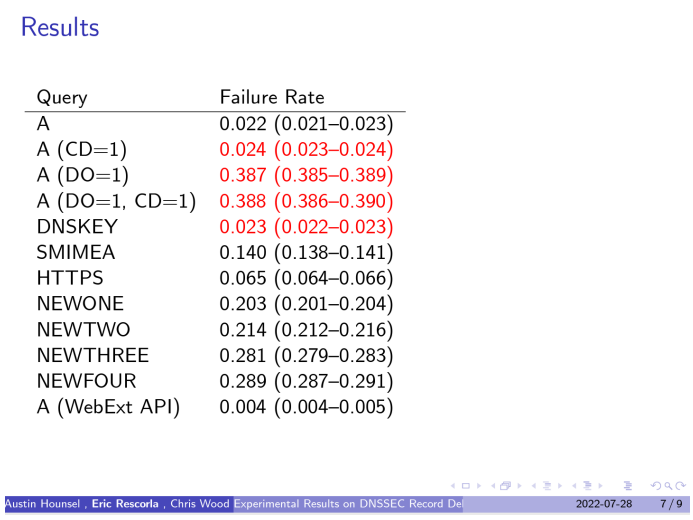
The document load in the DNSOP Working Group continues to be high, bordering on overwhelming. There are 36 non-working Group drafts that relate to aspects of the DNS and its operation in addition to the 17 active drafts and a further 6 drafts that have recently timed out and may be revived in one form or another. That's a very large volume of work!

As DNSSEC ages it accumulates more additional explanations, extensions and modifications and there is now a meta guide (draft-ietf-dnsop-dnssec-bcp) that essentially lists all the DNSSEC RFCS, of which there have been 33 to date. This process of incremental additions to the original specification leads to this plethora of specifications. From time to time, we hear calls to pull this together and integrate all the incremental elements into a single specification once more, but the task is sufficiently forbidding that all who have ventured into such a space emerge with their revising spirit broken! We are well down the path of writing "DNSSEC Good Housekeeping Guides" and the contribution draft-ietf-dnsop-dnssec-validator-requirements is a useful addition in this space. Most of these recommendations are already in the various DNSSEC RFCs so this document in the style of collecting already published information.

There is an interesting "dry run DNSSEC" proposal (draft-yorgos-dnsop-dry-run-dnssec). The idea is to allow validating resolvers to treat a signed domain as "under test" and if validation errors are encountered for the zone, then the resolver should treat the zone as insecure rather than failing with a validation error. Part of the acceptance issues for DNSSEC from the perspective of the zone publisher is that it's unforgiving, in that any errors in the management of keys and signature generation becomes a fatal error for the name. This flag allows a zone admin to test the entire publication chain without running the risk of an error taking zone offline for validating resolvers. (In fact, this is not entirely true. A counter case is the 2021 Slack DNS outage, where poor handling of the NSEC flags cause the zone to appear empty. Dry run would not help in such cases.)

One presentation was the highlight for me of this session and of the entire week. This was a report of work performed by researchers at Princeton University and folk from Mozilla and Cloudflare. The starting point was the observation that some 40% of users sit behind recursive resolvers that perform DNSSEC validation, and three quarters of these users, or 30% of the total seen set, exclusively sit behind validating recursive resolvers such they cannot successfully resolve a badly DNSSEC-signed name, yet almost none of these users perform DNSSEC validation themselves. The reliance on the recursive resolver for DNSSEC validation is close to complete. There are concerns about additional delays and breakage if the resolvers at the client-side were to start performing DNSSEC validation. There is also little in the way of real data to confirm, or otherwise dismiss, this concern. In this experiment the Firefox browser randomly enrolled clients to perform some DNS queries. The queries intentionally bypassed the local host platform DNS resolver library but otherwise did not avoid the local DNS resolution infrastructure. If the local network modem was configured as a forwarding resolver for the internal clients, then the Firefox browser passed its queries to this agent.

The queries resolved an A record via Firefox's dns.resolve() API. It also emitted queries with all combinations of the DNSSEC OK (DO) and CHECKING DISABLED (CD) flags. It queried for DNSKEY records, HTTPS records, SMIMEA records and small and large records in the Expert Review and Private Use ranges. Their results are shown in Figure 1. When the Query sets the DO bit the failure rate rises from 2% to 38%. The message appears to be quite simple: the customer edge of the Internet won't allow DNSSEC validation using undistinguished DNS queries across the edge DNS infrastructure. If we want to push DNSSEC validation all the way out to the applications then it seems that an encrypted tunnel, such as DoT, DoH or DoQ has a far better likelihood of success simply because it hides the DNS query from the edge DNS agents. It appears that these encryption techniques do more than alter the visibility of the user and their DNS activity. These approaches can also tunnel through broken edge network infrastructure.



Results

Query	Failure Rate
A	0.022 (0.021–0.023)
A (CD=1)	0.024 (0.023–0.024)
A (DO=1)	0.387 (0.385–0.389)
A (DO=1, CD=1)	0.388 (0.386–0.390)
DNSKEY	0.023 (0.022–0.023)
SMIMEA	0.140 (0.138–0.141)
HTTPS	0.065 (0.064–0.066)
NEWONE	0.203 (0.201–0.204)
NEWTWO	0.214 (0.212–0.216)
NEWTHREE	0.281 (0.279–0.283)
NEWFOUR	0.289 (0.287–0.291)
A (WebExt API)	0.004 (0.004–0.005)

Figure 1 – Failure rate of DNSSEC queries at the edge - from https://datatracker.ietf.org/meeting/114/materials/slides-114-dnsop-measuring-dnssec-success-01

SIDROP

The Working Group session on the operational aspects of RPKI and routing, SIDROPs, also met at IETF 114. There was an interesting presentation that surveys other PKIs and their methods of migrating relying parties to a new trust anchor. There are many approaches in use, and each have their issues. A "leap of faith" that substitutes a new trust anchor at the publication pint(s) used by the previous trust anchor are very vulnerable to various forms of substitution attack in a large distributed system. A more robust scheme is "old signs new" where the new trust anchor is deployed in advance, and some notice to this effect is published within the PKI, with a validation based on the old trust anchor. Following some period of such advanced notice, during which time the relying parties are expected to act on this notice and acquire the new trust anchor, the PKI can then switch to use the new TA. There are still some issues with this approach, and if a key isa ever hijacked, the attacker can then signal a trust anchor transition and move to a new key. In this case recovery is likely to involve a complete "stop and reset" of the system, which is, of course, highly disruptive.

The work with the roll of the key signing key in DNSSEC was guided by the advice contained in RFC5011, which mandated a lengthy stable announcement period. Trust in the new credentials was only established if the new material was stably published for this entire period. If the key had been hijacked, this extended time allows the legitimate key holder some time to detect the situation and to disrupt the attempted hijack.

This RFC5011 provisioning approach is being proposed for the RPKI trust anchor material, with a further addition that the successor key being proposed as the new trust anchor now includes a reference to the predecessor key, as an additional check to ensure that the successor key is configured correctly and is expecting to operate as a successor key.

There was also a discussion on the topic of delegated CAs and their use of repository publication points. The initial picture of deployment of RPKI credentials has largely relied on the so-called "hosted" model where subordinate CAs publish their signed products in a publication point repository operated by the parent CA. In practical terms this has meant that the majority of RPKI publication points are operated by the RIRs, which has some issues with scalability, robustness, and efficiency, and of course fate sharing. As the technology has become more widely deployed there has been some appreciation of the importance of reducing the potential for fate sharing by delegated CAs. This opens the options to use self-publication in dedicated publication points, or to avail themselves of the services of others, such as content distribution network platforms. Of course, this can go further. In the search for greater resiliency why shouldn't a CA avail themselves of the services of two or more third party providers for publication. It's always hard to determine the original intent of the authors of the specification here but was the original intent to allow for multiple publication points in a RPKI certificate that ability to define multiple access methods to the same repository, or to define a number of alternate repositories such that failure to complete an access again against one repository would allow a relying party to try others instead of simply giving up. The way that a CA migrates from one repository publication point to another is not specified in a standard manner.

There was an interesting discussion about the AS Provider Attestation and the interpretation of the Address Family field (IPv4 and/or IPv6). To my mind the underlying issue here is an issue of scaling. The RPKI is an instance of a space where all credentials need to be continuously accessible to all relying parties and the immediacy of the availability is one of the same timescale as the operation of the BGP protocol itself. If the design of the system makes each distinct signed object more specific than the population of objects managed through the RPKI increases. With a total of some 1.1M routed prefixes in the inter-domain space, and some 11,000 non-stub ASes out of pool of some 75,000 ASeS the scope of the encompassing object space is of the order of 1B. Now of course there is nowhere near that number of RPKI objects out there, but is there the visible outcome that we should encounter that many distinct RPKI objects, but there is a cautionary note to be aware of that taking design decisions that create highly granular objects heads further down this path of scale pressure than would otherwise be taken.

DINRG

In addition to Working Groups that are conventionally focussed on the production of standard specifications of technology there are a number of more open-ended groups that work within the overall IETF framework that tend to be more focussed on broader research into a particular topic. These so-called research groups are organised within the framework of the IRTF. The issue of centralisation in the Internet has been a topic that has grown in prominence in line with the growth of the small set of digital giants that have created this concern, so the Decentralised Internet Research Group has been s;un up to look at this issue.

In an effort to understand the causes of centralisation there was a workshop on this topic in 2021, and the meeting of DINRG at IETG 114 heard a report of the 2021 workshop content and themes. One view is that the current situation is not unique and there have been many similar instances in the past. The nature of the processes behind the industrial age and today in the digital age naturally favour larger players who can bring economies of scale to bear upon their competition and when a market swings from innovation to consolidation, aggregation is a natural outcome. This is further exacerbated by improvements in common infrastructure (canals, rails, roads, telephone, internet) that enable the projection of power and thereby increase the potential of scale through successively wider reach for the large players. So, in this context the internet is nothing special. But this is not entirely a simple repetition of the period of industrial expansion. These days the digital environment enables customisation at scale, so the more traditional delineation between mass market and customised goods has been destroyed and we are now able to produce customised goods at scale. The logical inference is that the volume incumbent is able to bring these economies of scale to all market sector, further increasing the market barriers to any new entrant.

At the same time, we are shifting the locus of control "up" the stack, devaluing the advantage of installed infrastructure and replacing it with software applications which require no coordination, no standards, and little in the way of overheads of working with others. So in some ways it's a similar story to the past, while in other ways the nature of centrality is larger and more pervasive and more of an issue.

Will the market "naturally" correct itself? This is unlikely, particularly when a monopoly is well established - competitors simply cannot gain a toehold. When public markets fail then governments are supposed to step in. So, we might ask: Will regulation help? We have enabled an abundant network that has enabled a global projection of power - national regulation is powerless and even regional regulation is visibly impotent. And in this particular case there is no regulatory template to follow. Also, just like the panic of 1910-1911 the use of regulatory instruments with due notice of the likely consequences will create uncertainty, and if the consequences are entirely unknowable then chances of inducing a larger economic depression are all the higher. Can users help? No - this business is focussed on giving the user exactly what they want, and most of the time they tend to be very good at adjusting their product and service to closely track shifting user tastes and demands.

This implies that the implied stasis of centralisation and monopoly will be with us for some time to come, no matter how we try to dismantle these central large scale dominant entities. For those who desperately want to believe in a future that is based around decentralised and ledger-less technologies, as typified in blockchain-like frameworks for example, this is not exactly a comforting message! And for those who think that technology can weave our way out of this situation, then for the near to medium term future all that this the historical perspective can tell us is that such an outcome is extremely unlikely.

IETF 114

This is a brief summary of a small set of topics from the IETF 114 meeting that attracted my interest. There were many more, including the world of IPv6, the Internet Area, routing topics, cryptography, security, operations and network management, and various research activities. The agenda for the meeting, including session notes, presentation material and links to recordings of the sessions can all be found at https://datatracker.ietf.org/meeting/114/agenda/

Disclaimer

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

Author

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

www.potaroo.net