February 2020
Geoff Huston

DNSSEC Validation (Revisited)

One year ago, I looked at the state of adoption of DNSSEC validation in DNS resolvers and the answer was not unreservedly optimistic (https://www.potaroo.net/ispcol/2019-03/dnssec.html). Instead of the "up and to the right" curves that show a momentum of adoption, there was a pronounced slowing down across 2017 and the first half of 2018 (Figure 1).

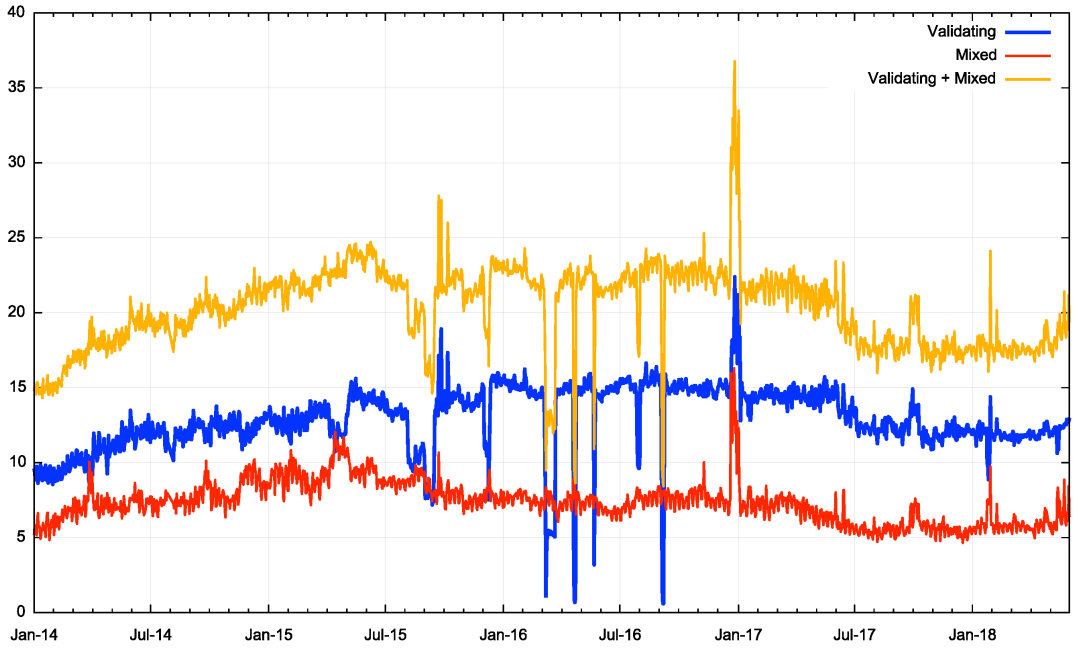

Figure 1 – DNSSEC Validation rate to July 2018

The current picture of DNSSEC adoption is certainly far more heartening, and I would like to update this earlier article on DNSSEC with more recent data. However, before going into the numbers, it's probably worth a brief recap of DNSSEC itself.

DNSSEC

Almost every transaction on the Internet starts with a query to the DNS. The DNS is the workhorse that translates the 'human friendly' names we use in URLs, email addresses and similar into IP addresses that allow applications to communicate with the named service. (Yes, the DNS is a whole lot more than this, but this one sentence summary is good enough here.)

Applications ask this amorphous service called *the DNS* questions about domain names and *the DNS* returns answers in the form of IP addresses. The question that DNSSEC tackles is: Can we trust the answer?

If I was able to intercept your DNS queries, I could substitute a different answer and redirect you to a different location altogether. The consequences might not be pretty!

This deliberate corruption of the DNS may sound like a fanciful notion, but it happens in many places in the Internet all of the time. Such DNS interception techniques are used for many reasons including state-based content censorship. A not so recent, but still informative, example of DNS substitution is the way in which access to the web site `www.facebook.com` is blocked within China https://www.potaroo.net/presentations/2013-08-29-facebook.pdf.

So how can we trust the DNS answer? Or, to put it another way, how can a DNS client tell if the DNS answer that they receive is a lie or not?

DNSSEC is the technology that can be used to answer this question. DNSSEC is an application of cryptography that attaches digital signatures to DNS records. These signatures can be validated by the user to determine if the record they receive is authentic or not.

DNSSEC has two parts: The zone administrator must generate cryptographic signatures on DNS records in a zone (or *sign the zone*). Secondly, a DNS client needs to check if the cryptographic signature is expected for this record, and that the signature record that they received is authentic. If so, then they can conclude that the DNS answer is indeed authentic.

What we are looking at here is DNSSEC Validation. The question we are trying to answer is "How many users use resolvers that perform DNSSEC validation to validate DNSSEC-signed responses?"

Measuring DNSSEC Validation

The measurement technique we are using here starts with an online advertisement. Ads are pretty much ubiquitous in the human-centric Internet, so if we want to conduct a large-scale sampling of the user base of the entire internet, ads are one way to achieve this.

Online ads are not just static images, but often contain HTML5 code. We use this code to task the user's browser to retrieve a collection of URLs. Care is taken to customize the domain name parts in these URLs such that every DNS name is unique. In this way we bypass any form of caching in the network, both in the DNS resolution phase and in the retrieval of the object identified by the URL. We also ensure that the only authoritative servers for this collection of DNS names are operated by us, as well as the matching HTML servers.

If the user successfully retrieves the URL then the DNS authoritative servers will first see one or more DNS queries for the name in question, and then the HTML servers will see a request for the named URL.

So that's the general framework used for such large-scale measurements. How do we measure DNSSEC validation?

In this case we use three URLs, each with a different DNS property. One is an unsigned domain name, acting as a control. The second is a validly signed domain name, located in a uniquely-named subdomain. The third is a domain name with an incorrect digital signature, also located in a uniquely named subdomain.

We look at both the DNS queries and the HTML queries. The experiment retrieves the control URL, retrieves the validly signed URL and does not attempt to retrieve the URL that uses the incorrectly signed URL, then this is a candidate user for using DNSSEC validation. There is, however one further

precondition used in this experiment. Because the subdomains are uniquely named, we expect to see DNSKEY and DS queries for this subdomain name as well.

If this is the case, then we conclude that all the recursive resolvers used by this user perform DNSSEC validation. Let's call this set of users **validating** users.

There is also a common case where a user has multiple recursive resolvers locally configured where only some of these recursive resolvers perform DNSSEC validation. When a DNSSEC-validating recursive resolver attempts to resolve an incorrectly signed DNS record, the recursive resolver will return the SERVFAIL error code and the local stub resolver will re-query using the next locally recursive resolver. In the case where only some recursive resolvers perform DNSSEC validation we will see the user retrieve the incorrectly signed URL, but also make DNSSEC resource record queries for the DNS name. Let's call this set of users **mixed** users.

When we add the case where none of the user's recursive resolvers make DNSSEC queries and the user retrieves all three URLs we can conclude that the user is not performing any DNSSEC validation whatsoever.

In this measurement experiment we count the number of users that we sort into each of these three cases. We are now in a position to look at the broader picture of DNSSEC validation.

The Internet-wide View of DNSSEC Validation

Figure 2 shows the complete measurement history from the start of this measurement in late 2013 to the end of February 2020. This time the plot shows the percent of **validating** users (blue) as well as the percentage of **mixed** users (red).

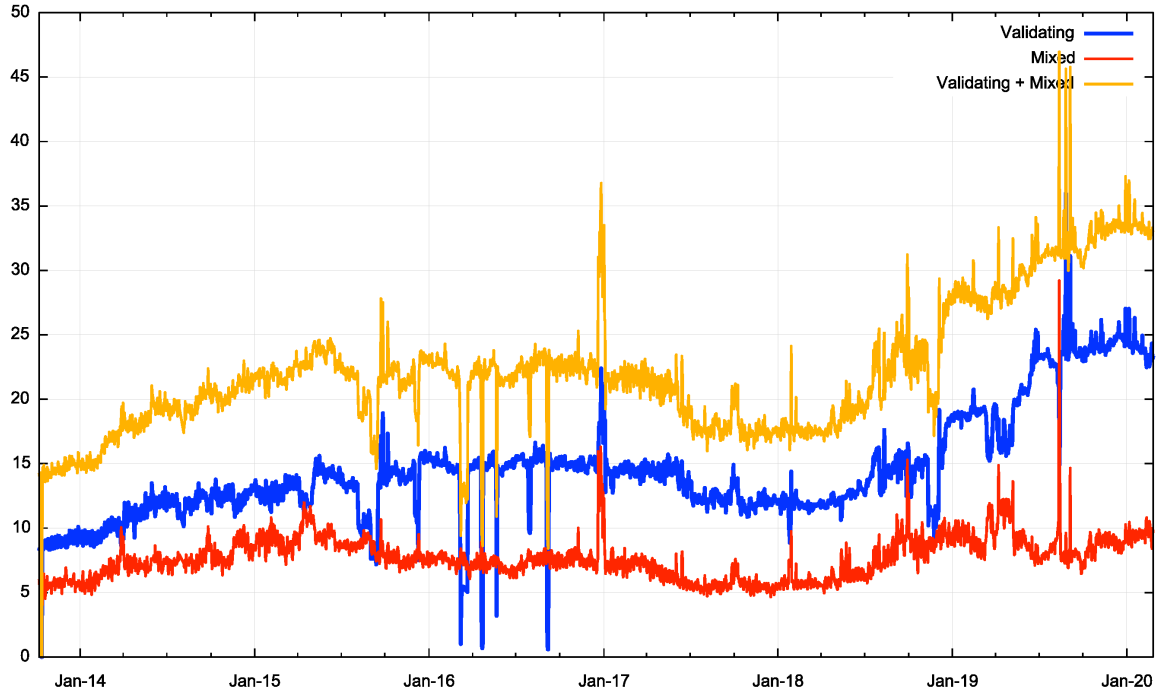

Figure 2 – DNSSEC Validation rate to February 2020

Since the second quarter of 2019 the population of **validating** users has risen from 12% to 22%, close to doubling. At the same time the proportion of **mixed** users has risen from 5% to 10%. Overall some 68% of users do not perform any DNSSEC validation at all at present, compared with 86% when this measurement started in 2013.

The DO Bit

The process used by a DNSSEC-validating resolver has two parts. The first part is to request that the response includes a digital signature (or RRSIG record) that can be used to validate the response, whether the response either contains data or is a NXDOMAIN (no such name exists). This response is loaded into the Additional section of the DNS response. The way this request is embedded in the DNS query is to use the extension options for DNS (EDNS(0)) in the query, and within that option set the DNSSEC OK bit. This bit requests the responding entity (recursive resolver or authoritative server) to include the RRSIG digital signature record, if the zone is DNSSEC-signed. The second part is the validating resolver will use this digital signature to validate the DNS response. This involves establishing a validation path of interlocking DS and DNSKEY resource records that form a chain of signing from the root zone Key Signing Key (KSK) to the Zone Signing Key (ZSK) used to sign the RRSIG record in the original response.

Interestingly many more users use resolvers include the DO bit sending in their queries than the number of users who use resolvers that perform DSNSSEC validation. While around one third (32%) of users sit behind recursive resolvers that perform DNSSEC validation (either mixed or full validating), some 90% of users sit behind recursive resolvers that set the DO bit!

While validation itself is a configuration option in many DNS resolvers, the use of EDNS(0) and the setting of the DO bit is evidently a default setting that cannot be readily altered by a local configuration setting. So, oddly enough, most of the DNS resolution infrastructure does the first half of the DNSSEC validation procedure. If a zone is DNSSEC-signed, then most users will generate DNS queries that include the DO bit set to 'on' and the RRSIG digital signature will be wrapped into the DNS response.

Why is this number of DNSSEC-validating users growing in recent months?

Some years back Google's public DNS service (commonly referred to as *8.8.8.8*) was a leader in the deployment of DNSSEC validation, as this service is widely used across the Internet. A growth in the use of Google's DNS service used to correspond with a growth of the level of DNSSEC validation. But does this still hold across the past two years? Figure 3 shows the DNSSEC validation rates as in Figure 2, but also adds the percentage of users who use Google's DNS service. It's clear that the rise in DNSSEC validation rates since January 2019 had very little to do with the use of Google's DNS service.

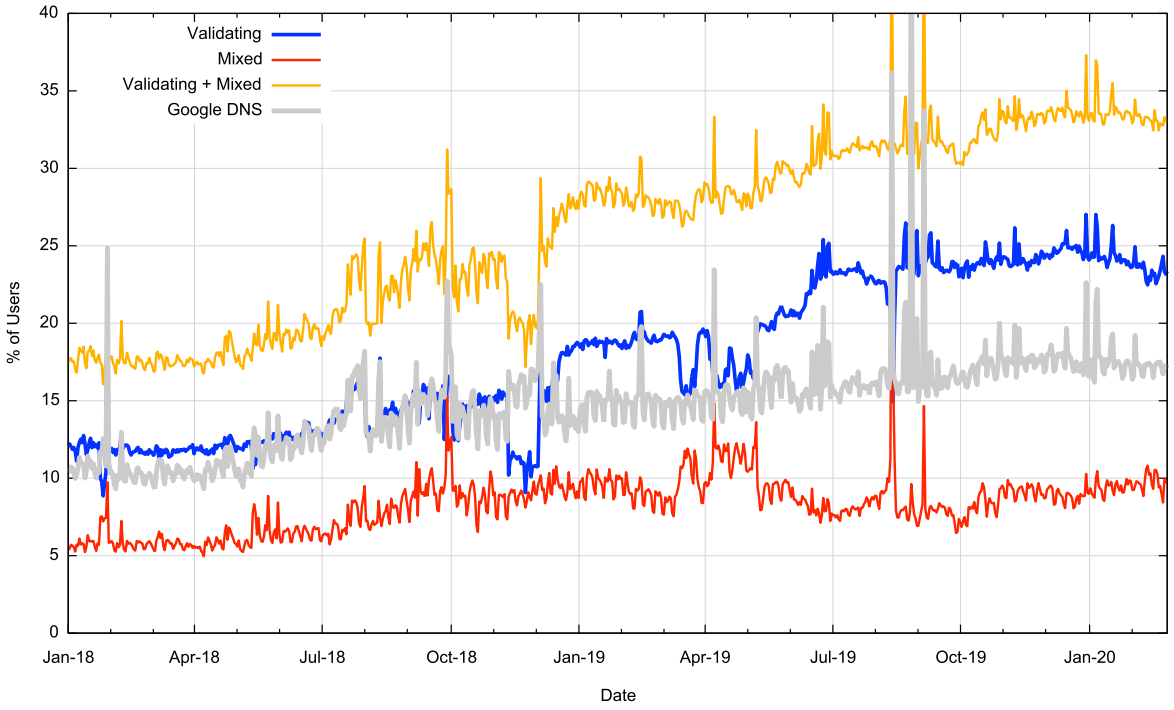

Figure 3 – DNSSEC Validation rate and use of Google's Public DNS Service

Where is this jump in DNSSEC validation rates coming from?

Regional Use of DNSSEC

Figures 4 through 8 show the same history of DNSSEC validation for each region.

The most significant change in DNSSEC validation since the start of 2019 is in Asia (Figure 7) and considering the significant population of this region, changes in Asia tend to have a significant influence on the global numbers.

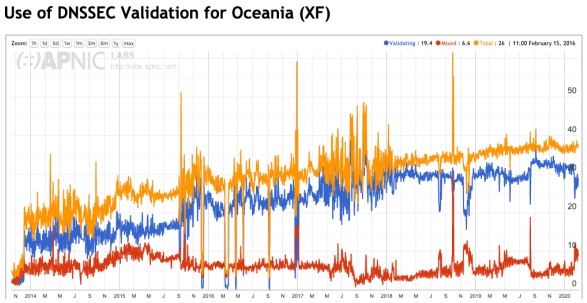

Figure 4 DNSSEC Validation, Oceania Region

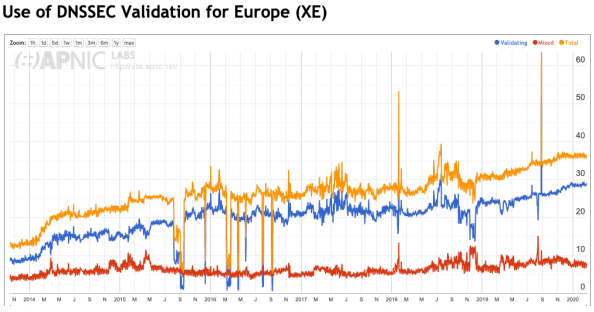

Figure 5 DNSSEC Validation, European Region

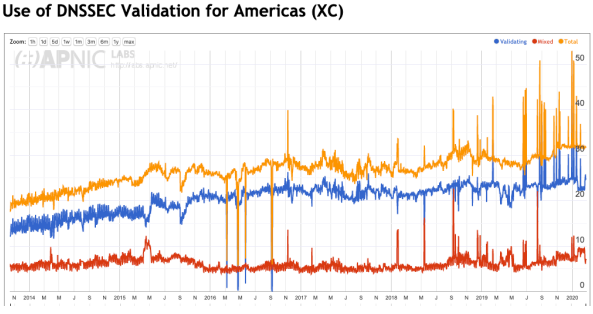

Figure 6 - DNSSEC Validation, American Region

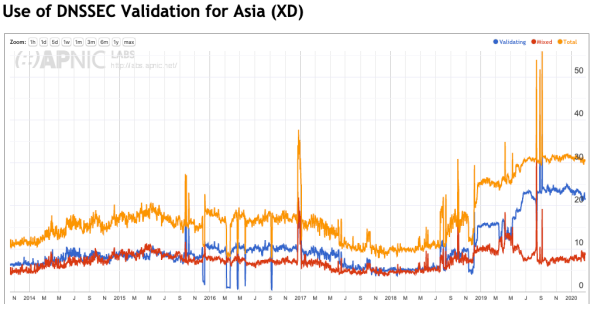

Figure 7 – DNSSEC Validation, Asian Region

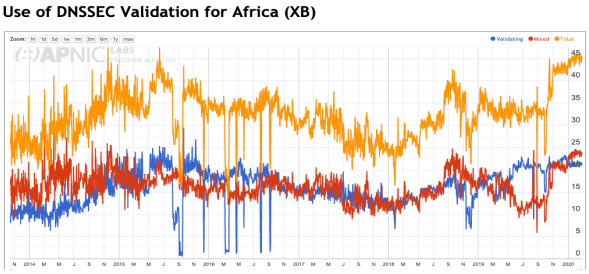

Figure 8 – DNSSEC Validation, African Region

But that's not the entire picture in the Internet service provider industry. The initial landscape of service providers was that or a large number of independent operators, with consumers spread relatively evenly between then within each national market. However, once the industry stabilised and entrepreneurial capital had quit the Internet service provider business, aggregation rapidly consolidated this environment, leaving a small number of dominant providers and a larger set of niche providers.

Mobile services have had a dramatic impact on this situation and the limitations of spectrum availability has resulted in a typical outcome of three major mobile service providers and a far smaller set of resellers within most economies. Here the pressures for provider aggregation have been present from the start. The result of these pressures is that there are today just 16 service providers that provide Internet access for one third of the global Internet user population. The point here is that when any of these major providers in this group of 16 decide to turn on DNSSEC validation in their recursive resolvers the global DNSSEC adoption numbers take a visible jump upward.

In 2019 2 of these very large providers make such a change to their DNS services. In India, the mobile provider Reliance Jio, clearly India's largest ISP, made the decision to enable DNSSEC validation in late 2018, completing the work by mid-2019 (Figure 9).

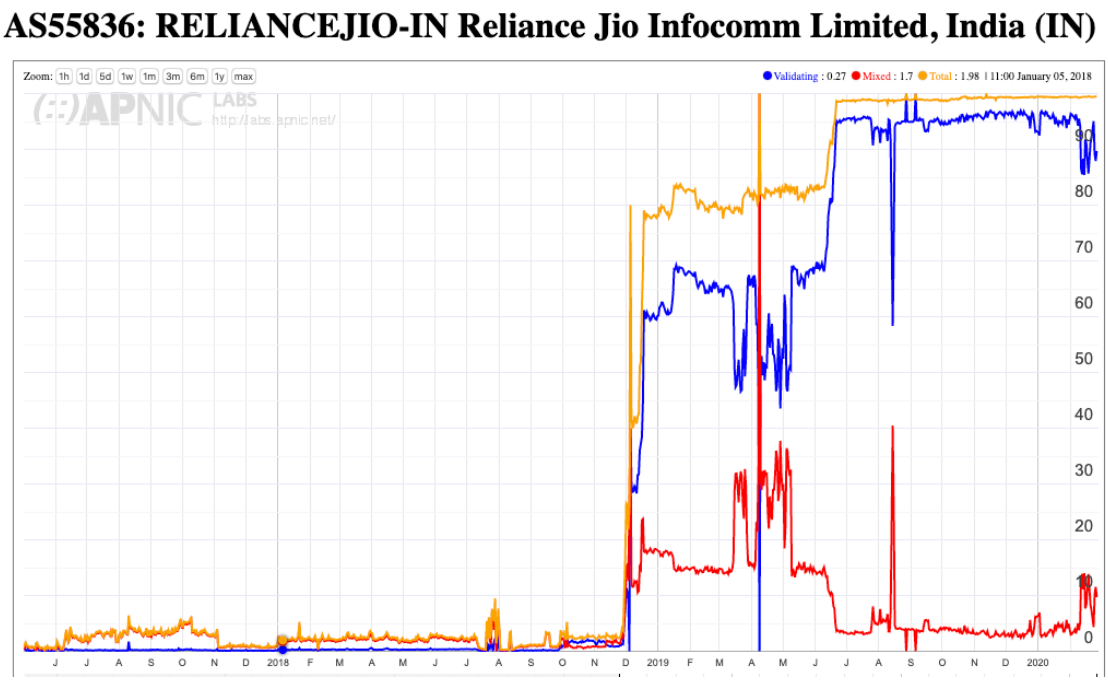

Figure 9 – DNSSEC validation for AS55836 (Reliance Jio)

In Germany the major national provider, DTAG, enabled DNSSEC at the start of 2019 (Figure 10).

AS3320: DTAG Internet service provider operations, Germany (DE)

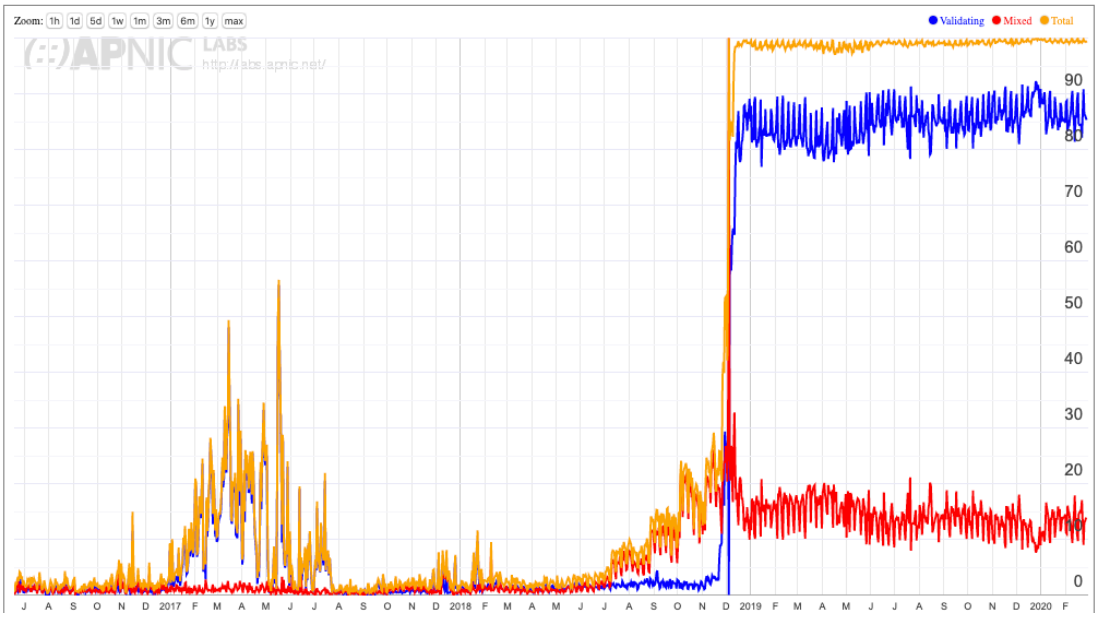

Figure 10 – DNSSEC validation for AS3320 (DTAG)

Why the hiatus in DNSSEC Validation in 2017 and 2018?

It's probably not coincidental that the roll of the root zone Key-Signing Key was originally scheduled to occur in October 2017. A significant effort was put into pushing out a message that DNSSEC-aware DNS resolvers needed to be tracking the roll of the KSK, and if they failed to follow the progress of the incoming key then the resolver would fail to answer any queries. The diagnostic tools for resolvers to determine if they are accurately tracking the KSK trust state are pretty woeful, and it is a challenge even for operators of these resolvers to get the resolver to report on its trusted key state. It is quite conceivable that a number of resolver operators took the conservative option in 2017 and turned off DNSSEC validation as a way of avoiding potential service problems with the KSK roll.

There are no substantiated reports from these service operators as to why DNSSEC validation was switched off in 2017 and why it was resumed late in 2018, so the linking of this data with the KSK roll is simply supposition on my part. But in my opinion the hiatus in DNSSEC validation across the entire Internet was directly related to the extended period of the KSK roll. It seems entirely logical that many operators who were contemplating turning on validation in their resolver infrastructure delayed the process until the KSK had been rolled and DNSSEC was once more considered to be "stable".

But this raises a more fundamental question about this technology. If operators feel that it is entirely reasonable to defer, or even turn off, a technology for more than a year, then why turn it on at all? If it's it not perceived as being a necessary or vital part of an ISP's DNS service portfolio then why bother doing it at all?

Let's take a deeper look at exactly this question.

Is DNSSEC worth it?

It seems that there is much uncertainty over the use of DNSSEC validation. Some resolver operators appear to have embraced DNSSEC and use it as a point of principle. This set of DNS operators includes the large open resolver networks operated by Google and Cloudflare. On the other hand, there are resolver operators who do not perform DNSSEC validation. While DNSSEC validation is now being used by some 25% of the world's users, some 65% of users direct their queries to DNS resolvers that do not perform DNSSEC validation in any form at all.

Who's right? Is DNSSEC validation a good idea? Or is it just a whole lot of effort with little in the way of tangible benefit?

There is no single answer to this question. DNSSEC offers a more resilient and trustable DNS where users can trust that the DNS answers that they receive exactly match the authoritative zone contents. But this comes at a cost, and the issue is whether the benefits are worth the incremental costs of adding DNSSEC signatures in DNS zones and validating these signatures in DNS responses. Let's look at both sides of the issue.

The Case for "No!"

It's easy to see DNSSEC as a case of one more thing to go wrong in the DNS. For DNS zone administrators it's another element to the zone administration tasks, adding key management, regular key updates, zone signing, key rollover, and coordination of keys with the parent zone and delegated zones. Even the simple elements of zone delegation and zone contents are mis-configured in much of the DNS and adding the elements of cryptographic keys and digital signatures only adds to the probability of zone failure for those who choose to sign their zone.

DNSSEC adds to the size of DNS responses, and this creates potential issues with the DNS. DNS over UDP is meant to fit responses within 512 bytes. Adding DNSSEC digital signatures generally causes the response size to exceed this limit. DNS queriers need to use EDNS extensions to specify their capability to handle large UDP responses and for larger responses there is the issue of IP packet fragmentation and the subtle differences between IPv4 and IPv6 in terms of IP packet fragmentation. The backstop of reversion to TCP adds additional time and additional unreliability as DNS over TCP is not universally supported in all DNS resolvers. But perhaps this is a not quite as big a concern as this commentary would leave you to believe. I've already noted that some 90% of users sit behind recursive resolvers that set the DO bit in their queries, and the inclusion of the DNSSEC digital signature in DNS responses appears to be the default mode of operation for today's Internet. If adding a RRSIG record to responses presented a significant operational problem for the network, then we would've encountered the issues of problems with large DNS responses already. The lack of any such broad evidence points to a conclusion that this is not a major issue for the DNS.

DNSSEC validation takes additional time. The need to assemble the DNSKEY and DS records of all of the parent zones without resolver caching would present an insurmountable time penalty. The use of resolver caches mitigates this additional penalty in resolution time, as these resource records span an entire zone. For example, one would expect a DNSSEC validating resolver to retrieve and cache the DS and DNSKEY RRs for .com and .net zones almost immediately. However, in a world where every millisecond matters DNSSEC is an extravagant time waste!

Most end systems do not perform DNSSEC validation directly. They rely on their DNS resolver to perform DNSSEC validation on their behalf, and they implicitly trust in the resolver to perform this with appropriate levels of integrity. But in the dialogue between the stub resolver and the recursive resolver the recursive uses a single bit to indicate that DNSSEC validation was performed and this response was validated (the AD bit). Of course, the issue here is that a man-in-the-middle attack between the stub resolver and the validating resolver is still potentially effective: the end host is not validating the DNS response and cannot detect if a response is genuine or if it has been tampered with. Security relates to the entirety of a system and the weakest link principle applies. [Allowing the recursive resolver to validate the responses form authoritative servers is good, but passing this as a single unprotected bit in the response to the stub resolver remains a glaring weakness. Stub resolvers should perform DNSSEC-validation directly in a secured environment. But that raises its own questions about the viability of the DNS to scale, as adding DNSSEC validation to every stub resolver adds both delay and load to the DNS.

DNSSEC validation outcome error signalling is inefficient. Validation failure re-uses the SERVFAIL response code, which acts as an invitation to the recursive resolver to spend more time performing re-queries using different authoritative servers for the zone (which, admittedly, is an improvement on an earlier behaviour when the resolver would exhaustively check every possible delegation path!).

DNSSEC validation is variable. When and how do resolvers perform validation? Do they perform all the queries for DNSKEY and DS records before attempting validation? Do they serialize these DNS queries or perform them in parallel? What about CNAME records? Is the first name validated before following the CNAME or is the CNAME record followed and then both names validated? How do these additional tasks and the time taken to complete them interact with existing timers in the DNS? It appears that DNSSEC causes additional query load in the DNS because of this interaction between aggressive timers in the client and time to complete validation functions that need to be performed by resolvers.

Of course, there is also the really tough question: What threat is DNSSEC protecting you against? The textbook answer is to protect resolvers against the so-called "Kaminsky attack" that injects bad data into a recursive recover's cache. DNSSEC can certainly provide this protection, but only in a limited context. It can protect the recursive resolver, but the non-validating client stub is still as vulnerable as ever. Therefore, this is not a comprehensive solution to the problem. It's a step in the direction of threat mitigation by potentially protecting recursive resolvers against man-in-the middle attacks. But is the cost of this DNSSEC response commensurate with the nature of the threat? This particular DNS attack appears to be a rather esoteric attack vector and the use of randomised source ports in resolvers already adds sufficient randomness to make the Kaminsky guessing attack somewhat ineffective in any case.

The overall impression from this perspective is that DNSSEC is half-cooked and the costs far outweigh the potential benefit of risk mitigation.

The Case for "Yes!"

The overall picture of security in the Internet is pretty woeful. The path between recognising a URL on a screen and clicking on it and believing that the presented result is actually the genuine article requires a fair amount of blind trust. We are trusting that the DNS mapping of the name to an IP address is genuine, trusting that the routing system is passing the IP packets to the 'correct' endpoint, trusting that the representation of the name on your screen is actually the name of the service you intended to go to, trusting that the TLS connection is genuine, and trusting that the WEB PKI is not corrupted, to name but a few.

That's a lot of trust and many would argue it's just too much trust. As we place more and more personal and social functions into a world of connected computers, we place more and more reliance on the integrity of the Internet. If an adversary can subvert the Internet's functions, then there is considerable potential for disruption and damage. The experience from repeated attacks so far is that adversaries, whether its talented hackers, criminal enterprises or state actors, can subvert the Internet's operation and infrastructure and can create considerable damage. And with the much-touted Internet of Things on the way we are about to over-populate this already compromised environment with even more devices, and place even greater levels of reliance on a foundation that is simply incapable of withstanding the pressure.

There is no single cure here and no single measure that will make it all better for the Internet's infrastructure. We need to improve the resilience of the addressing and routing infrastructure and we need to harden the DNS and make it more resistant to attempts to compromise it. DNSSEC may be a work in progress but as far as the DNS is concerned DNSSEC is all we have.

But let's not underestimate the value of a robust trustable name infrastructure. The trust infrastructure of the Web is repeatedly exposed to the corruptible and the efforts on certificate transparency, HPKP records and CAA records are desperate and largely ineffectual measures that fail even when using a limited objective of palliative mitigation. If we want to provide essential web security then it appears that domain keys in the DNS are the best response we have to the issue (DANE), and that means we need to have a trustable DNS where users can verify DNS data as being authentic. We are now upping the stakes with the use of encrypted SNI in TLS 1.3. Encrypting the SNI field in the TLS handshake removes the last piece of overtly visible data on the wire that shows the end point intended destination name. But the problem is how to communicate the SNI encryption key to the user in a secure manner. Here a combination of DNSSEC and DNS over TLS or HTTPS (despite its potential for circularity) might be

the only viable way through. The client performs a DNS query to retrieve the ESNI key over a secured connection, and then performs DNSSEC validation on the response to assure itself that the ENSI key value is authentic. At that point the client can encrypt the SNI value and instantiate a TLS 1.3 connection using a ESNI field value. DNSSEC appears to be the only way we know how to achieve this outcome.

Work is going on with addressing some of the more obvious shortcomings of DNSSEC validated name resolution. A shift to use elliptical curve cryptography can reduce the size of digital signatures and reduce DNS packet sizes to avoid some of the issues related to IP packet fragmentation. The use of DNSSEC chained responses could improve the efficiency of DNSSEC validation, but at the expense of larger responses, which implies that such a move to use DNSSEC chain extensions in responses would make a lot of sense in a context of DNS-over-TLS or DNS-over-HTTPs, or as a stapled attribute in a TLS certificate exchange to facilitate the use of DANE as a CA-pinning measure. A refinement of the DNS error codes to explicitly signal DNSSEC validation failure would prevent the resolver re-query behaviour that we see with SERVFAIL signalling. Work is also underway to equip end hosts with DNSSEC validation capability, so that end hosts are not reliant on an untrusted (and vulnerable) connection between the host and their DNS resolver. And let's not forget that caching in the DNS is incredibly effective. The digital signatures in DNSSEC are cached in the same way as delegation and address records are held in the cache, so there will be no real time penalty for validated resolution of a signed DNS name if the relevant resource records are already held in the local cache.

Without a secure and trustable name infrastructure for the Internet, the prospects for the Internet look pretty bleak. DNSSEC is not the complete answer here, but it sure looks as if it's an essential element of a secure and trustable Internet. And maybe that's sufficient reason for us to adopt it. We can and should put in the technical effort to make DNSSEC more efficient and make it easier and faster to use. But we shouldn't let the perfect be the enemy of the good. There is no point in waiting for a "better" DNSSEC. We'll be waiting indefinitely, and the problems associated with a compromised digital infrastructure will persist. The alternative is to simply use what we have at hand with DNSSEC and use our ongoing experience to shape our further efforts to harden up both the DNS and the larger Internet infrastructure.

For securing the DNS there is no Plan B beyond DNSSEC. Any story relating to improving the security of the Internet necessarily entails securing the name system and that inevitably involves the use of DNSSEC.

The overall impression from this perspective is that DNSSEC is already deployable. But "deployable" is not the same as "completed", and the work is not finished by any means. Operational experience will guide the further refinement of DNSSEC tools and techniques.

.

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