The ISP Column
A monthly column on things Internet

January 2008

Geoff Huston

DNSSEC – Once More, with Feeling!

After looking at the state of DNSSEC in some detail a little over a year ago in 2006, I've been intending to come back to DNSSEC to see if anything has changed, for better or worse, in the intervening period.

Background

To recap, DNSSEC is an approach to adding some "security" into the DNS. The underlying motivation here is that the DNS represents a rather obvious gaping hole in the overall security picture of the Internet, although its by no means the only rather significant vulnerability in the entire system. One of the more effective methods of a convert attack in this space is to attack at the level of the DNS by inserting fake responses in place of the actual DNS response. The outcome can be quite insidious. You may think that you have pointed your browser at some web site, but if I can execute this attack on your DNS query then it would be possible to direct your browser to any address whatsoever, or even none at all! And nothing is obviously "wrong". There are no viruses on your computer, no failure of your firewall, no insidious subversion of your NAT. Nothing is changed at your end and everything is working just as intended. But the wrong outcome happened. And of course the DNS is used as the universal rendezvous medium, so pretty much every application starts by firing off a DNS request and then commencing a packet handshake with whatever IP address the DNS has told it. With DNS as it stands today such attacks on the integrity of the DNS, particularly by quite simple man-in-the-middle substitution attacks, are virtually impossible to detect.

DNSSEC is a digital signature framework that is intended to allow anyone to check the integrity and completeness of a response to a DNS query. The approach used by DNSSEC is a relatively conventional application of public key cryptography, where DNS Resource Records (RRs) are digitally signed. This digital signature can be added as additional information to a DNS response, and DNSSEC-aware resolvers can request that additional DNSSEC validation material be provided in addition to any DNS query, and the resolver may then use this material to verify that the response is genuine, and has not been altered in any way whatsoever. For a description of the DNSEEC design you may find a previous article on DNSSEC helpful, namely "DNSSEC: The Theory" http://www.potaroo.net/ispcol/2006-08/dnssec.html.

DNSSEC deliberately avoids the use of X.509 public key certificates and the use of an associated Public Key Infrastructure (PKI) by embedding a key hierarchy within the DNS itself. Each zone parent is given the role of signing over every delegated child's zone key, so that a zone's signing key can be verified by confirming the parent zone's signature over that key, which, in turn, can be verified by that zone parent's signature over this key, and so on until you reach a Trust Anchor or the root of the DNS, or, preferably both at once. All this results in the observation that the theoretical trust model of DNSSEC can be reduced to a reliance on trust in only one key, that key being at the top of this delegation hierarchy. In other words, DNSSEC was originally designed with a single trust anchor in mind, that being the public key used to sign the root zone of the DNS, and the deployment model was one of universal adoption across the entire public DNS.

So where are we on this DNSSEC deployment agenda? Within reach? Or a bit of a stretch goal, but still plausible? Or maybe its so far out there that the manned mission to Pluto will happen first!

Root Signing

Some tasks appear to be quite challenging, yet turn out to be really quite simple, while other tasks look simple, but turn out to expose all kinds of daunting problems. In the case of DNSSEC this task of

generating a public/private key pair, signing the root zone with the private key and publishing the public key as the material that "anchors" the DNSSEC signing hierarchy is evidently an example of this latter class of tasks, where the seemingly simple becomes the intractably wedged. The root zone of the DNS is quite small, and generating a key pair and creating a digital signature of the root zone is not actually the problem. Indeed, it's a mechanical task at one level and existing DNSSEC tools can achieve this outcome quite easily. An example of what such a signed zone might look like can be found at https://ns.iana.org/dnssec/status.html.

If the mechanics of actually signing the root are so trivial, and we know what the outcome will look like, and you can probably even use the DNS to test your resolver against it tomorrow, then why hasn't it happened already on the actual DNS root? What's the problem here?

Obviously its not technical. So if its not technical, then it must be back to yet another instantiation of the those endless political debates about the DNS, IANA, ICANN and the roles and desires of various governments, political factions of all persuasions and their various retinues and cheer squads. What appears to have got so many folk worked up into a lather in the case of DNSSEC is the consideration of ***who*** generates the key pair for the DNS root. Actually that's not quite correct – the observed general paranoia levels start to go stratospheric over the consideration of who has effective control of the private key of the DNS root zone, as this is evidently equated to the well rehearsed public debate that has a paranoia level that could best be described as astronomical in elevation over who exercises actual control over the DNS root zone itself, and somehow this gets even further inflated into the fascinating debate of who has control over the entire universe or something similarly grand.

Descending back to the boringly mundane for a second, there has been a proliferation in number of folk who believe that they have some legitimacy in their claim to have a say over control of their little part of the DNS root zone and perhaps an equally large number of folk who are interested in disputing the comparable claims of others. So who gets to hold this magical key of the DNS root appears to be a Deep and Meaningful Question for many.

I must admit that for me the most obvious response to such DNSSEC root zone signing concerns goes along the lines of: "What's the issue here? The IANA maintains the DNS root zone file, so the IANA should sign the root zone and the IANA should control the zone keys for the DNS root."

While such a response may appear obvious to me, for many that response is totally naive politically and just plain unacceptable. We've seen a number of proposals in recent times that attempt to justify the employment of a cast of agencies drawn from many countries and many international organizations, each of whom are proposed to derive their rasion d'être in being vested the control a tiny part of one bit of the DNS root key, and all of whom are required to provide their consent in order to assemble the actual key value to sign any changes to the DNS root zone. Somehow the mechanical process of signing parts of the DNS root, which is a task that is required for every change to any part of the root zone, so we are talking of an operation that entails a number of signing operations per day, appears to have been confounded with the political process of an endless debate about the process of making changes to the DNS root. I gather that there is the impression in some circles here that there is the endorsement of a cause that a functional outcome can be achieved when every single party here in this somewhat bewildering cast of players has the right of veto over anyone else. The key that would allow others to bootstrap the validation hierarchy in a simple manner is now regarded as the key to define which parties need to provide their permission before permitting any changes whatsoever to the DNS root zone itself. For as long as the evident desire on the part of some to ensure that this political process of DNS determinism is non-terminating holds prominence in the public debate over the governance of the DNS, then the prospects of ever seeing a signed root zone in the DNS also appear to be highly unlikely.

But if one were to dismiss most of this debate of who gets to say which bit of the root zone key should be a 1 or a 0 as a simple exercise in political posturing, and look at this merely as a technical problem within the DNS query and response protocol, the more relevant question is whether adding a DNSSEC signature to the root zone of the DNS would cause any major, or even minor, catastrophe to the Internet at large, or the working of the DNS in particular.

The general consideration when contemplating changes to the root zone of the DNS is whether the basic UDP response to a root zone priming query will be able to be transmitted back to the query party. When a priming query is passed towards a root server with the DNSSEC OK bit set in the EDNS header of the DNS the query should also use the ENDS "sender's UDP payload" mechanism, as the inclusion of the DNSSEC information in the priming response will lift the size of the responding DNS packet to over 576 bytes. So as long as everything between the DNSSEC and EDNS-aware DNS entity performing the query who is asking the root zone the priming query with DNSSEC enabled is capable of doing the right thing with large UDP packets, then all will work. And if the packet is mangled by some reprehensible middleware that mangles DNS UDP packets in bizarre ways, then the query has to drop back out of

DNSSEC and perform a non-DNSSEC priming query. It seems that the operational modes do not cause complete breakage here. So as long as the key is correctly handled and the signatures are well formed then from a technical perspective signing the root does not appear to present risks to the operation of the DNS.

Signing the DNS root appears to remain a political question rather than a technical question. For as long as there are folk who equate their unwavering desire to express their interest in the politics of the administration of the DNS with an undeniable conviction that they or others deserve a right of veto in the administration of the root zone of the DNS via their interest in a share of the control of the DNS root key, then this may well remain an intractable political problem. Sigh.

Signing Everything Else

There is one arguably positive aspect of this longstanding political stalemate over signing the root zone of the DNS, namely that the other rather improbable aspect of DNSSEC deployment can be conveniently ignored.

DNSSEC relies on the DNS structure itself to create the interlocking relationships that remove the need for additional verification mechanisms. For a DNS RR to be verified by a DNSSEC-aware resolver it is necessary to both verify the digital signature of the RR and confirm the validity of the signing key that generated the RRSIG resource record. In DNSSEC the parent signs across the child's public key, so the child's key can be verified by verifying the parent's digital signature of this key. This signature can be verified by confirming the digital signature and verifying the parent's public key. This key is verified by its parent, and so on. If the root zone key is the trust anchor of this verification chain then every zone between the root zone and the zone of the DNS RR being verified must be DNSSEC signed. With DNSSEC it's a case that life is far simpler for clients of DNSSEC if over on the signing side of DNSSEC it's a case of one in, all in!

Such a perspective poses the question of just how likely is universal DNSSEC deployment? The story is not yet very convincing, unfortunately. In many cases it appears that the marginal additional overhead of adding DNSSEC to a zone, in terms of the additional zone administrative procedures, the introduction of DNSSEC key management functions, larger DNS zone files, interactions with delegated zone administrators and parent zone administrators, and the possibility of more potential points of service failure, are all borne by the zone administrator ands the zone publisher, while the direct benefits, such as there is direct benefit here, accrue to the DNS client. If we've learned anything about the business of networking over the years it would be good to think that we've learned that in those situations where a large set of folk bear the costs and another, potentially smaller number of folk stand to gain the benefit, and there is no compensatory flow of money to even the scorecard, then the entire proposition is not generally considered overly attractive from a business perspective. Worse still is the picture where the sum of the costs far exceeds the accumulated value of the benefit. Signing everything in the DNS just doesn't seem to represent a natural outcome from the point of looking at each DNS zone administrator as an independent entity who is trying to optimise their own individual position on the cost and benefit account.

Universal DNSSEC deployment, even with a signed root, has prospects that are even more dismal than the signed root proposition. Sigh

The Great Key Hunt

So we haven't yet managed to sign the root, and we haven't yet managed to achieved universal buy-in below the root, and the prospects for achieving both these objectives look dim at the moment.

Are we ready to give up on DNSSEC yet? Of course not!

The design of secure systems often represents a set of design trade-offs, or, in other words, a set of compromises between security, scalability and feasibility. If it is the case that the total cost of deployment and the resultant accumulated benefit are not well aligned, then can DNSSEC take some convenient shortcuts to ease this imbalance, even at the expense of some level of integrity of the security model? Is there any leeway here? Is universal DNSSEC adoption a strict precursor to any realistic level of DNSSEC utility?

In the absence of universal DNSSEC adoption then we might continue along the current path of partial adaption for some time to come. But what does this mean for DNSSEC? Is partial deployment of DNSSEC something that we could live with in the long term? Or is it more broken than no DNSSEC at all?

Partial deployment of DNSSEC implies that only some zones are signed. A signed zone might have no DNSSEC parent, but may have DNSSEC children. These parent-less DNSSEC "orphans" become the apex of a local DNSSEC hierarchy. For a DNSSEC resolver to be able to verify as much as it possibly can it has to load up all the zone keys that are at the apex of a local DNSSEC hierarchy. Unfortunately there is no automated way to perform such a sweep of the DNS to expose these DNSSEC zones, so the process appears to be one that you perform by hand. Yes, that's manual. And even then you need to derive some form of trust in the authenticity in the key in this undefined process of DNSSEC zone key retrieval and maintenance.

Another way for the resolver to gather these zone keys up is to pick them up one by one on as as-required basis using the DNS itself. This gets rid of the entire process of trying to sweep the DNS for these DNSSEC local hierarchies just in case you might want to pose a query against these zones that you wanted to validate with DNSSEC. So it looks promising. But hang on – wasn't the entire DNSSEC effort directed at providing some means of verifying DNS responses? If you use the DNS itself to deliver these keys that you are going to use as the basis of your trusted verification, then haven't you just introduced a fatal circularity of dependence? How can you trust the DNS to deliver you the correct key that you are then going to trust to validate DNS responses? So this won't work either.

So we are back once more to the basic problem of DNS integrity that DNSSEC was intended to solve. How can you pick up these local DNSSEC apex keys in a reliable and trusted manner without resorting to the DNS? Unfortunately DNSSEC has no direct answer here. How do you know that a DNS zone has been legitimately signed and that the DNS response can be validated? In an environment of partial DNSSEC deployment DNSSEC does not appear to be overly helpful.

Another response has been to contemplate DNSSEC Lookaside Vectors (DLV), which attempt to aggregate a number of apex zone keys of local DNSSEC zone hierarchies into a synthesised DNSSEC zone that allows a single DLV zone key to substitute for the set of stored apex zone keys. If you change the DNSSEC-aware resolver to follow the lookaside vector you can replace the collection of local apex keys with the single key of the lookaside vector zone, as long as all these DNSSEC orphans are prepared to live as the DNSSEC Lookaside Orphage. But in this DLV model we've broken out of the interlocking DNS delegation model, and the question then arises as to the authenticity of the zone keys stored in this DLV zone. While DLV can fly technically, the validity of the outcome is no longer based on the elegant structure of interlocking DNSSEC keys that are precisely aligned to DNS zone delegation. Instead, the strength of the outcome is no stronger than the integrity and accuracy of the admission procedures that are undertaken by the DLV operator. Its hard to see how the DLV approach offers anything more secure than the current haphazard collection of DNS name certificates and the haphazard collection of certificate authorities who issue certificates for DNS names, while at the same time sit outside the name delegation hierarchy.

So the alternatives for the DNSSEC client in a world of partial DNSSEC deployment is to hunt down and maintain a set of trust keys using undefined processes that presumably involve a considerable amount of human direction, or to outsource the problem to a DLV operator in whom you are placing trust that they operate their DLV business with suitable integrity.

Neither alternative sounds overly attractive if you are after the substance of security rather than the optimistically inclined veneer of security theatre. Sigh.

No Such Domain

But even if we were in a world of universal DNSSEC deployment the story still is not overly convincing. If DNSSEC means signing every response that the DNS can offer, then the DNS response of "no such domain" must also be signed, and this too has opened up some further twisted DNS byways.

The DNSSEC mechanism of signing such negative responses appears to be both simultaneously ingenious and flawed. Its ingenious in that the approach of generating pseudo DNS RRs of all the "gaps" in the zone file and signing these "gap" DNS RRs provides the appropriate assurance that there is indeed no such domain in the zone in a manner that allows verification of this absence of requested information. Its flawed in that the approach not only provides information that the particular domain name does not exist, but also provides extraneous information about a potentially large set of other names that also do not exist in the zone file.

In any other domain (if you'll pardon the pun!) any side effects of this DNSSEC signed NSEC resource record would be innocuous, but in the strange and twisted world of the name selling business information about what names do not exist appears to be a highly valued asset. NSEC is just too revealing a response

for many players in the domain name industry. What NSEC does is to place an additional RR in the zone and for each name in the zone the RR contains the next name in order, using a standard ascii ordering of the names in the zone. The obvious inference is that all the names that lie between the two names don't exist, and the result is an indirect method of zone enumeration.

So its been a case back to the drawing board to devise an alternate approach of signing these gaps in the zone file. From this comes the still-being-cooked approach of NSEC3, which uses a far more complex ordering of the records in the zone than simple ascii ordering. NSEC3, which is still not yet signed off as a stable specification, does not appear to have made DNSSEC any easier or simpler. Indeed quite the opposite. Forbidding and incomprehensible appear to relate to both the design goals of NSEC3 and the outcome! I suppose top marks are in order to the NSEC3 designers for having achieved their objective. What is not so clear is whether this has succeeded in making DNSSEC forbiddingly complex. Sigh.

The DNSSEC Burden

DNSSEC is not free. The DNSSEC burden is spread across the zone administration, primary zone servers, secondary servers and resolvers and the end clients of the DNS. Way back – way way back in the late 1980's the DNS traffic on the NSFNET contribute 20% of all packets on the network of the day. Obviously things have improved considerably since then (we hope!) and its not anticipated that DNSSEC deployment is going to break either the DNS or the Internet at large, but, even so, DNSSEC is not free.

For the zone administrator there's the additional tasks of key management, record signing and managing zone updates. For the primary and secondary zone servers there's the need to support incremental updates to the zone, the advisability of using a trusted channel for zone updates from the primary server to the secondaries, such as TSIG, the larger zone sizes, the larger response sizes and the corresponding increments in processor, memory and bandwidth requirements for the servers. This, in turn, triggers reconfiguration of platform and network capacity for the server side of the DNS.

For resolvers there is the issue of the larger response sizes, the need to ensure that resolvers and the local network infrastructure of firewalls, NAT boxes and other inventive pieces of middleware can cope with EDNS0 and larger DNS response packets that are potentially fragmented, the additional query overhead associated with validation of responses, and that tricky question of what DNSSEC outcomes to cache and for how long.

One major consideration here is that the DNS is already relatively fragile and is the constant target of attack. One way to disrupt a service is to subject its name servers to a constant very high intensity load. In amongst all this noise traffic genuine queries are lost and the servers effectively disappear off the network. For providers this is a balancing act. While the DNS is essentially unfunded, if you are out of action then the outage is highly visible. Given that the attack has been one of loading the DNS server with correctly formatted queries there is no visible clue to the server as to what queries are genuine and what constitute the attack. The common mitigation to this attack is firstly one of segmentation of the server domain through anycast of the DNS servers, and an effort to increase the capacity of the server to absorb the query load associated with an attack without falling over. In this environment we are now seeing implementations of customised DNS servers and recursive servers that are engineered for very high capacity. In this environment DNSSEC applies more pressure through larger responses. A DNS server that is engineered for resiliency in a non-DNSSEC world may not necessarily be able to manage the increased load is all queries in a concentrated attack had the DNSSEC bit included and the zones being used as the target of the query were DNSSEC signed. From the perspective of a large heavily used DNS zone, DNSSEC represents a likely requirement to reinvest in server infrastructure as a consequence of DNSSEC signing the zone. This is not exactly delivering on the generic promise of all this technology as being better, faster and cheaper. Sigh.

What's the meaning of "failure", and who cares anyway?

What should a resolver do in the event of a DNSSEC verification failure?

Lets look at this in a little more detail. If the DNS response is a RR and the associated DNSEC RR signature fails to verify then what should the DNS resolver present to its client? "Server Failure" is just such a pathetic way of backing away from the problem without taking a stance! But if a DNSSEC aware resolver doesn't want to appear to be such a pathetic indecisive wimp, then should the resolver synthesise a NXDOMAIN response in place of the suspect RR response, on the basis that passing on a response that has failed verification is probably worse than a "no such domain" response? What if the NSEC (or NSEC3 for that matter) response has a DNSSEC RR signature that cannot be verified? Here the "server failure" response is still relatively unhelpful and but a more assertive NXDOMAIN, or "no such

domain" response is possibly an incorrect response, but, equally there is no better information at hand to substitute in its place. There is nothing quite like the quandary you are p[aced in when trying to do the "right" thing when you know that the answer you are about to give to your client is shonky, but being completely unable to do anything about it!

But what is failure anyway? And, more particularly, what is "failure" in a partially deployed DNSSEC world? A digital signature that cannot be verified is a clear failure. The lack of an upward chain of parent signatures that leads to a trust anchor is also a failure, but of a different type. It could well be that in this case the DNS RR provided as the answer to the original query is perfectly good in every respect, including the DNSSEC RR signature and it's the resolver's efforts to hunt down all of the apex zone keys for each and every local DNSSEC hierarchy. Or the resolver could be the victim of a DNS attack. How can a resolver tell the difference? Is this a benign failure, or an instance of a failure that points to a directed attack via the DNS.

Of course undermining all this is human behaviour. When the screen presents you with the message: "I cannot validate this certificate, do you want to proceed anyway: Yes or No?" we are all pretty much the same when we move the mouse to hover over that "yes" button. DNSSEC might be as paranoid as it wants, but we humans are all risk takers of one sort or another, and even when presented with a dire warning of the risks involved, there's a certain air of digital bravado that creeps up upon your clicking finger when given the opportunity to take a risk! So all this security infrastructure can be undermined by an all too typical user behaviour mode. Sigh.

So tell me again…

It still seems somewhat bizarre to me that much of the public consideration of DNSSEC has been on the topic of the number and composition of the group of people who must have their trembling fingers touching the pen that signs the root zone of the DNS. The politicization of this question of "who signs the root?" appears to ensure that the root remains unsigned, and as a consequence any hope of universal deployment of DNSSEC flies out the window. It appears that without a signed root partial deployment of DNSSEC is the best one can hope for, and partial deployment of DNSSEC is, on the whole, an instance of negative progress.

Without a signed root DNS resolvers need to do the impossible and perform minor miracles in terms of trust key discovery and management. The impossible happens only with considerable pain and effort, and even then it happens rarely. The question is then raised as to why should zone administrators and DNS operators incur additional costs to locally deploy DNSSEC given that there are a vanishingly small number of DNSSEC queries to be answered from these few bold and adventurous DNSSEC-aware resolvers? From this perspective it appears to be counter-productive to be fixating on the issue of how many potential signatories should be dancing on the head of this root zone pin .

Surely the answer to signing the DNS root is one that has been staring us in the face since the start of this entire DNSEC effort. As with all zones, it the role of the zone administrator to generate the keys and sign the zone file. In the case of the root zone of the DNS its back to the IANA to just do the job and let the rest of us move on.

But "moving on" is not the same as solving all the issues of Internet insecurity though the scribbling of just one signature over a block of bits with a digital pen. It is apparent that the lack of a signed DNS root has become a convenient way to paper over the other issue of the lack of DNSSEC deployment across the remainder of the DNS and the poor prospects for ever changing that situation. Its still a very hard problem to work out how to swing the balance around in the DNSSEC cost and benefit equation so that the benefits of deploying DNSSEC on the server side can motivate its deployment such that the client side benefits of a more robust DNS can be realised.

And even if we address that and move into a DNSSEC world, there is the observation that most of the technology failures we encounter in this area are outcomes of benign rather than hostile actions, and that failures are all too often a product of some failing of operational competence rather than intended malice, and this is no doubt going to continue. We have become habituated as seeing "failure" as an accidental and unintended outcome, rather than a dire warning of potential danger. We have grown so trusting in the technology that when the technology offers us choices then we believe that it really is a choice between equally viable options and that both options are equally 'safe' from the point of view of the underlying technical machinery. Otherwise why would this machine be asking us for guidance in the first place?

The substantive issues for DNSSEC are much further down in the DNS hierarchy than at the root, but we're never even going to have the opportunity to address them as anything more than hypothetical issues to considered in the abstract for as long as the unsigned DNS root remains in our way.

Disclaimer

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

About the Author

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001.

http://www.potaroo.net