The ISP Column
An occasional column on things Internet

October 2006

Geoff Huston

DNSSEC - The Opinion

Over the past two articles we've looked at the theory and practice of DNSSEC. So is DNSSEC really going to happen? Is this useful technology that will establish itself as part of the DNS infrastructure? Or is the technology fatally flawed in some fashion that will lead to its premature demise? This final article in this examination of DNSSEC will enter into areas of conjecture and opinion over the future of DNSSEC.

DNSSEC has been around for some time. The earliest internet draft I could find on the topic is one written in February 1994 (draft-ietf-dnssec-secext-00.txt) as part of the work of the DNS Security Working Group of the IETF, and there appears to have been some activity in this space in 1993. So the concepts behind DNSSEC have been in circulation for the past 13 years now. However it appears that it has been only in the past couple of years, with the refinement of the DNSSEC model and the evolution of the network and end system capabilities that DNSSEC looks to be readily deployable.

So if DNSSEC is now readily deployable why haven't we deployed DNSSEC already? Is DNSSEC really hanging by a thread, and is it really a case of "now or never", as a relatively recent Internet T-Shirt by Olaf Kolkman suggests?

Lets look at the various perceptions of DNSSEC and see if we can add to the already hefty set of opinions and conjectures about DNSSEC.

DNSSEC – The Negatives

What appears to be the common negative perceptions of DNSSEC?

- DNSSEC creates bloated DNS zone files.

  Don't forget that over the same dozen years that DNSSEC has been developed the Internet has grown from megabit trunk systems to gigabit trunk systems, and end systems have similarly increased in processing and capacity metrics by a similar three orders of magnitude. By comparison the DNS zone file inflation with DNSSEC is much more modest. The example zone I signed grew in size from 843 bytes to 7379, or less than one order of magnitude. This is not really a case of 'bloat'. Zone files are larger, with DNSSEC, but not prohibitively so.

- DNSSEC generates higher levels of DNS traffic.

Some years back, in 1989 as I recall, the DNS accounted for over 20% of the NSFnet backbone traffic. Since then the DNS story appears to have only improved! DNS traffic is a relatively insignificant proportion of today's networks, and compared to massive flow of various forms of multimedia traffic, DNS fades into invisibility. The additional traffic that could be attributed to universal DNSSEC deployment simply is not an issue today.

There is, however, one aspect of DNSSEC on the wire that is a significant concern. Any UDP-based protocol that takes as input one small trigger packet and produces in response one or more large packets directed to the (unverified) source address can be perverted to act as a traffic amplifier. DNS servers and resolvers have been enrolled as unwitting accomplices in denial of service attacks simply because of the property that a small DNS query produces a larger DNS response. With DNSSEC its certainly the case that the same small query produces a much larger DNSSEC response. The combination here that makes the attack effective is one of the ability to perform source address spoofing and udp-based transaction protocols where the response is significantly larger than the query.

- DNSSEC is complicated

  Well, yes, managing a signed zone is certainly more complex than not. To quote from the "DNS and Bind" book (4th Edition)

  > "We realize that DNSSEC is a bit, er, daunting (We nearly fainted the first time we saw it). But its designed to do something very important: make DNS spoofing much, much harder. As people do more and more business over the Internet , knowing you're really getting where you thought you were going becomes crucial."

  I suspect that its not really the complexity of public key cryptography that is the heart of the complaint. I suspect that its not even the toolset itself that's the problem. My personal suspicion is that it's the state of the documentation and supporting material that makes this a daunting task. Many novel tasks where there is uncertainty and confusion appear complex to start with.

- DNSSEC increases the workload of zone administrators

  True. Private keys need to be protected. Public keys need to be rolled over. All changes to the zone file need to be signed with the zone key. Zone integrity in terms of verification of the zone's contents needs to be monitored. (see draft-ietf-dnsop-dnssec-operational-practices for a more complete laundry list of tasks for the DNSSEC zone administrator) These are certainly additional tasks for the zone administrator. And the busier the zone in terms of adds, removals and changes then more work this entails.

- DNSSEC exposes more information about the zone

  Ah – the NSEC / NSEC3 argument again! Yes, NSEC records in a zone implicitly allow a third party to enumerate all the entries in a zone file through zone-walking along the NSEC values. This is not a failure of DNSSEC per se. DNSSEC needs to provide some information that would allow the relying party to validate a negative (no such name) response. The failure here is a failure of DNSSEC to align to the emerging business models of DNS name registrars. The technology response is to use a more convoluted canonical ordering of the zone file with NSEC3 records, so that there is a limited amount of zone information exposed in an NSEC3 response. Whether this approach allays these commercial concerns over information exposure, or whether the NSEC3 records are seen as one step too far in terms of algorithmic complexity remain to be seen.

- DNSSEC has a poor trust model

  I can't agree with all of this, but there is a gem of trust in this particular assertion. DNSSEC would have a perfectly fine (and very robust) trust model if your zone parent signed your key fingerprint, for each and every DNS zone. And to avoid the implicit reference to unterminated recursion here (this reference to unterminated recursion brings to mind the phrase "Its turtles all the way down" [http://en.wikipedia.org/wiki/Turtles_all_the_way_down]) all a relying party would require to terminate validation is a local copy of the DNS root key. In this way validation of a signed DNS object becomes validation of a key, and this, in turn, is an exercise of establishing a

sequence of validly signed key fingerprints that lead to the DNS trust anchor. However, we are not in an environment of comprehensive DNSSEC deployment, and the partial deployment we have today is very troublesome. How do I know if the absence of DNSSEC additional information in a DNS response to a query is a result of a deliberate attack on my query or simply that the zone is unsigned? When you sign your zone, and your zone parent is not DNSSEC capable, then how can I decide whether or not to add your keys to my trusted key set? After all if the zone parent does not sign the child's zone, then the binding of the child zone keys to the child zone is, in effect, an untestable assertion of zone control that relying parties have little choice but to simply trust if they choose to use these keys. If I choose not to add this key to my trusted key set, then how can I tell the difference between a zone that does not validate because of a key failure (bad) and a zone that does not validate because my trusted key set is incomplete (possibly not so bad)? My experience with DNS lookaside vectors (DLVs) does not appear to help here. The DLV model works outside the normal delegation model so the issue of establishing the bona fides of clients of the DLV service are problematical and costly. Consequently, the DLV space is sparsely populated and there is little incentive for resolvers to use it. So its not that DNSSEC has a poor trust model. It appears to be that partial deployment of DNSSEC creates a somewhat confused and confusing trust model.

- DNSSEC represents a large investment for a marginal return

  Hmm. That's a tough one. Today, for a large zone operator DNSSEC represents a considerable outlay, or investment in DNSSEC. And given the paucity of clients who are in a position to use the outcomes in a positive way, and given he implicit trust anchor weaknesses in a partial DNSSEC deployment, the incremental value that DNSSEC would add to a DNS resolution transaction is small. So, perhaps like IPv6 today, DNSSEC represents an investment where the returns will only be evident in the fullness of time, and the returns are strongly conditional on everyone else making the same investment over time.

DNSSEC – The Positives

So why bother with DNSSEC? Because having DNSSEC is far better than not, in terms of the network's integrity. The Internet is not a network-centric communications architecture. The network itself is not the privileged possessor of user's credentials, nor does it perpetuate a model of limitation transactions with end devices that attempts to protect its own integrity. In the Internet's architecture the network loses this distinguished role and undertakes a far more basic role of commodity packet switching. In consequence, users are placed in the position of establishing new forms of mutual trust that are, at times, completely unfounded and very vulnerable to deliberate efforts of subversion.

There are perhaps two deep areas of trust on which the integrity of the entire network relies upon – routing and the domain name system. And in both cases the associated information distribution model is unauthenticated, and in both cases it is vulnerable to various attack as a result. In the case of the DNS the problem lies in DNS spoofing, where a query receives a response which has been tampered with by the attacker. In and of itself this is not a significant issue, but when the application uses this false information to create a session and undertake transactions, then the risk factors tend to multiply.

- DNSSEC can provide a robust form of defence against DNS spoofing attacks

  Because DNSSEC allows the DNS client to authenticate the received response against a credential framework, DNS spoofing becomes a whole lot harder, as the attacker needs to not only alter the DNS response, but also pervert the client's validation of the credentials associated with the response.

- DNSSEC provides a platform for authenticatable information distribution

  Jakob Schlyter has pointed me towards OpenSSH and its used of DNSSEC in RFC 4255, "Using DNS to securely Publish Secure Shell (SSH) Key Fingerprints". To quote from the document's abstract:

> "This document describes a method of verifying Secure Shell (SSH) host keys using Domain Name System Security (DNSSEC). The document defines a new DNS resource record that contains a standard SSH key fingerprint."

I suspect that this is but one of many ideas that use the DNSSEC property that information that can be bound to a domain name can be authenticated against the domain name, through the use of DNSSEC.

- DNSSEC exploits the delegation hierarchy of the DNS itself

  DNSSEC is not an unnatural imposition of trust that cuts across the natural delegation hierarchy of the DNS (In stark contrast, might I add, to the DNS certificate model that underpins SSL and the entire https framework, where the certification model appears to be completely arbitrary, or at least based on considerations other than explicit mutual recognition of a trust relationship. Just look at the collection of Trusted Root Certification Authorities that have been preloaded into your browser, and ask yourself if you have ever heard of all of these folk, let alone want to trust them with your assets!). In contrast DNSSEC has the potential to simply exploit the existing process of zone delegation, and make the zone delegation explicit through the explicitly verifiable action of signed key fingerprints. Assuming a comprehensive model of DNSSEC deployment where all child zone key fingerprints are signed by the parent zone and published in the DNS, then the trust model is based on the DNS root as the privileged trust anchor.

DNSSEC - Some Personal Opinions

Despite the obvious potential of DNSSEC to be a positive contribution to the Internet in its role of public communications utility. There is something that has been deeply unsatisfying about the exercise I documented in the previous column in an attempt to set up DNSSEC-signed zones and a DNSSEC-aware local resolver, and perhaps this is due to the related observation that, for me, there is something deeply unsatisfying about the state of DNSSEC today.

I suspect that the original vision of DNSSEC was, and still is, a sound and worthy one – I suspect that the original technology vision was a relatively lightweight mechanism of authentication of DNS responses that would allow DNS users (the you and I of the Internet world that use the DNS to drive browsers, mail, and almost every other application) to be able to test that the DNS responses that they get are "genuine" responses that have not been tampered with. Certainly the DNS is a major point of vulnerability for the Internet, and this, hopefully simple, mechanism of DNS authentication has the potential to make some significant progress in risk mitigation within the DNS.

But somewhere in the past 10 years of the DNSSEC development effort we seem to have deviated somewhat from this overall direction and admitted some further complexities into the model that are unhelpful at best. It seems that the issue over the exposure of zone contents through the implicit walk-though of NSEC records and the consequent development of the rather complex NSEC3 framework is one that has nothing to do with users, and way too much to do with the commercial concerns of the domain name supply industry. Zone enumeration is not an implicit security weakness – it's a business issue over information exposure. The fundamentals in authentication is that some level of information has to be exposed to allow a relying party to perform adequate authentication actions. To deliberately occlude the contents of a zone file in order to make the negative information response required for authentication essentially a meaningless one does appear to be a triumph of "we can do that" over "we should do that".

While NSEC3 is an interesting case study of computational complexity imposition, the situation with respect to trust injection in DNSSEC is perhaps the more dire aspect of DNSSEC today. The continued lack of a signed root, and the lack of signed top level domains, has turned the task of configuring DNSSEC resolvers into a rather frustrating game of trusted key hide and seek, a game that really does not work for operators of DNS resolvers. The response of working on lookaside vector approaches relating to trust key aggregation do not appear to represent a viable replacement to the original model of a comprehensively signed DNS infrastructure, and appear to me as ad hoc responses with poorly constructed trust models that can do no better than the https certificate efforts in terms of robust authenticatable trust.

Maybe its another aspect of the politicization (and commercialization) of the DNS that has occurred over the past decade. The DNS is no longer primarily motivated by users who want reliable resolution of name strings, but by name registrars and domain servers who have highly focussed commercial interests that, obviously, have considerable sway over the DNS. If we want to make DNSSEC simple and useful then we really want to avoid haphazard trust injection into DNSSEC

and, preferably, we'd really like to use a comprehensive DNSSEC deployment framework, from the root down. After all this is all about allowing end users to validate the authenticity of information that they receive via the DNS. The question DNSSEC is attempting to answer, on behalf of the end user, the resolving party, is: "Is this precisely the same information that was placed there by the DNS zone manager?" A comprehensive DNSSEC structure allows this question to be posed with a minimum of overhead on the part of the end user. But the acceptance of a comprehensive DNSSEC framework appears to hinge on the determination of those with a stake in the DNS root to proceed with the signing of this root zone. It also implies that the next level down, that of the various top level domains operators, that they also perceive sufficient commercial value in DNSSEC deployment that would at the very least offset the incremental costs associated with its deployment, and, preferably, generate handsome profits from such a DNSSEC investment. Obviously, given the relative paucity of DNSSEC deployment right now, this is not happening at present.

Where we are today, with small-scale partial deployment of DNSSEC and apparently random placement of associated points of trust, is not a useful blueprint for long-lived global deployment of technology. DNSSEC technology may be perfectly fine (although I'm still personally somewhat sceptical of NSEC3 records), but the associated DNSSEC trust model we have wandered in to today is potentially a fatal flaw for DNSSEC. Today the pain and cost of adoption by servers and users is not balanced by the benefits of having a mechanism to potentially validate a subset of DNS responses. Without a clear rendezvous mechanism between DNS publisher and DNS consumer, then even if you, as a DNS publisher go to the trouble (and cost) to sign your zone and all its descendants, then how can I, as a DNS consumer, figure out that you have done this? Where are your current trust keys? And I should trust them or not? As a DNS consumer my enduring value proposition lies in the ability for applications that I run to be able to apply a validation test to each and every DNS response. This implies that each and every DNS zone publisher sees value in signing their zone and maintaining current keys with their zone parents all the way up to the root. For DNS zone publishers, the enduring value of DNSSEC lies in an outcome where all resolvers chose to validate responses as a preference, and are unwilling to use unvalidatable DNS responses under any circumstances.

The current impasse has the appearance of a mutual deadlock situation, where the supplier and the consumer remain immobile while they wait for the other party to make a move. The longer term problem with such a situation is that after spending some years in one of these pre-deployment stasis situations the technology itself stands the risk of being labelled as a commercial failure, irrespective of its actual merits or potential utility in the future.

I suspect that DNSSEC sits in this form of stasis right now. It's not a problem looking for a solution. It is indeed a reasonable solution for a known problem (well, I'm not completely sure about NSEC3 when making this claim, but the rest of DNSSEC looks logically consistent to me!), and can make other problems easier to address as a beneficial side-effect. The situation with DNSSEC appears to be one of making this particular solution easy, simple, cheap and useful to deploy, and easy, simple, cheap and valuable to use.

The true leverage of DNSSEC is perhaps only realized in retrospect – if we get to the position where all servers and all resolvers are using this technology, then the value of DNSSEC in protecting the integrity of the DNS in an efficient manner would probably be clearly evident to all. But getting from here to there appears to involve a path of incremental steps that I'm not sure that either the DNS "producer" or DNS "consumer" is really willing to follow right now. How to identify natural incentives for more DNS "producers" and DNS "consumers" to go through the same exercise of zone signing and response validation appears to be an important near-term goal in the path to DNSSEC deployment.

Personally I would be pleased to see DNSSEC head off into widespread, indeed ubiquitous, deployment – the Internet would be all the better if this were to happen. But I'm still not sure that I can see a clear path from where we are today to where DNSSEC needs to be in terms of deployment. We'll see.

Further Reading:

http://www.dnssec-deployment.org

A somewhat varied (I'm tempted to say "eclectic") set of materials relating to the deployment of DNSSEC. For example, I found out from this website that NSEC3 already has its own domain name and web site (http://www.nsec3.org) and that there was a workshop on NSEC3 held on 18 – 20 September 2006. A quick pass through this material still didn't provide me with any decent leads as to how to make the trust key management in an environment of piecemeal DNSSEC deployment any easier for resolvers.

http://www.rssac.org

Why isn't the root of the DNS signed? This web page of the dns root servers advisory committee (RSSAC) appears to suggest in their meeting minutes that the revised schedule for signing of the root is some three months away – always! There are also some indirect pointers to the differing schools of DNSSEC deployment, namely the "bottom-uppers" and the "top-downers."

CircleID – DNS Topics

http://www.circleid.com/community/topics/view/Domain%20Name%20System/
I find CircleID to be a decent read, and the DNS forum contains a number of articles that relate to DNSSEC, including some perspectives on the interaction between political and commercial considerations and DNSSEC deployment, particular in terms of signing the DNS root (always a popular topic!)

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre, nor those of the Internet Society.

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.