Realities in DNSSEC Deployment

Paul Hoffman

DNSSEC was expected to usher in a new era where answers to DNS queries could be relied on as cryptographically authentic and thus be used to leverage other Internet transactions in a secure fashion. That new era has yet to happen even though the protocols are fifteen years old and the DNS root has been signed for over eight years.

If DNSSEC were as widely deployed as had been expected, follow-on protocols which require DNSSEC would allow relying directly on the DNS, instead of third parties like those used in the Web, to authenticate data such as name-to-address mappings. However, those add-on protocols have languished because DNSSEC adoption has languished.

Lack of Signing

For DNSSEC (as described in RFCs 4033-4035) to be considered widely deployed, a large percentage of the most-used domain names in the DNS would need to be signed. Many people have encouraged widespread signing by name owners, but with only limited success.

In many countries, very few of the most popular web sites sign their domain names. Top sites like google.com, weibo.com, and facebook.com do not sign their zones (although yandex.com does). Even companies that understand how to sign with DNSSEC are hesitant to sign their most popular names. Of the more than 550 “dot brand” new gTLDs (all of which must sign with DNSSEC due to contractual obligations), only about 5% also sign their associated .com names.

When questioned about why they do not sign, zone owners often cite protocol complexity and lack of reliability of signing software, although many large enterprises reliably sign their zones. Whenever strict authentication is used, the possibility of bad signing (such as not re-signing before a signature expires, or of signing over the wrong items) means that the signing software must be highly reliable, and in the case of DNSSEC the risk of mis-signing may be considered too high relative to the benefit that DNSSEC gives.

Another reason sometimes given for not signing is the high cost of hardware security modules (HSMs). However, nothing in the DNSSEC protocol requires the use of HSMs, and many large zones do not use HSMs at all. The perception that HSMs are required may come from the fact that the root zone is signed using HSMs in widely-publicized ceremonies; if this is a deterrent to signing, it is an example of highly-visible parties using best practices scaring “normal” protocol users.

Lack of Validating

Ongoing testing performed by APNIC indicates that about 17% of Internet users use a resolver that does DNSSEC validation, and trend analysis show no indication that adoption rates are

going up. The same tests show that about 15% of Internet users rely on Google DNS (which does DNSSEC validation) for their resolution.

The latter fact should give pause to those who are thinking about the effects of service centralization in the Internet. If Google DNS's users instead were behind other typical resolvers, or if Google DNS stopped validating, the number of users behind validating resolvers would drop well below 5%, into the "negligible" range.

Although there are complaints about complexities and difficulties in DNSSEC signing, such complaints are rarely heard about using DNSSEC validation in resolvers. For most resolver software, turning on DNSSEC validation involves changing just one or two lines of the main configuration (and some distributions come with DNSSEC validation turned on by default). Still, it is hard to find enough ISPs who validate to push the measurements of validation other than by Google DNS high enough to call DNSSEC an implementation success.

Lack of Last-Mile Authentication

When a stub resolver (commonly run on a computer as part of its operating system) sends a query to validating resolver over normal DNS on port 53, the answer comes back with no authentication. The response has a bit in the header (called the "AD" bit) set to 1 to indicate that the validating resolver believes that the answer passes DNSSEC validation tests, but the message itself is not authenticated.

Unlike the problems of lack of signing and lack of validation, lack of authentication to the user was largely due to lack of protocols that would give that authentication. The DNSSEC protocol in 2005 assumed that if an end system needed authenticated answers, it had to authenticate them itself by having the trust anchor for the DNS root, receiving and caching all the DNSSEC information for each query it made, and validating responses itself. It was not until DNS-over-TLS was published in 2016 that there was a standardized way for a stub resolver to use channel security to be sure it was getting authentic answers from its resolver, and then be able to rely on the semantics of the AD bit.

Conclusions

DNSSEC is not alone among cryptographic security protocols: S/MIME and OpenPGP have been around longer and also have adoption rates that are well less than was expected by the communities that created them. The primary problem with missing the expectations of the designers of cryptographic security technologies is that deployment of protocols that rely on widespread adoption of those security technologies is then blocked. A subsequent problem is that proposals for security technologies that might be more widely implemented may also be blocked due to lack of clear reasons why the first technology failed to be adopted.