Closing the DNS Security Loop with DNSSEC

Peter Losher
Internet Systems Consortium
FreeBSD Developer Summit
May 11, 2010 / Ottawa, Canada

Table of Contents

What is DNSEC?

DNSSEC (**D**omain **N**ame **S**ystem **SEC**urity Extensions) is a set of IETF (Internet Engineering Task Force) extensions for DNS which give DNS resolvers/clients the ability to verify the origin of any DNS response. DNSSEC was designed to protect Internet resolvers/clients from forged DNS data from events such as DNS cache poisoning.

What it does:

"The DNS security extensions provide origin authentication and integrity protection for DNS data, as well as a means of public key distribution" (RFC 4033)

What it doesn't do:

"These extensions do not provide confidentiality." (RFC 4033)

What is DNSEC?

Oh and by the way...

While you are doing this, be backward compatible with the current DNS framework.

Why do we need DNSSEC?

DNS as it stands now is a very insecure protocol and is very trusting. RFC 3833 describes in detail how a local DNS setup (mostly resolvers) can be compromised, the typical method being “cache poisoning”.

“Cache poisoning” is a method of inserting a forged response to a DNS query by a local resolver. Since the local resolver caches that data (for performance reasons), the local cache is considered “poisoned”.

Over the past decade DNS implementors have tried to make it harder and harder for successful cache poisoning attacks (port randomization, etc.) but we are running out of bullets from the current gun...

How does DNSSEC work exactly for the end user?

DNSSEC uses public key cryptography to digitally sign answers to DNS requests.

Under DNSSEC all answers are digitally signed with RRSIG (**R**esource **R**ecord **SIG**nature) record so all queries to a signed zone under a DNSSEC aware server will serve the original request, plus the RRSIG record associated with the original query.

A resolver that is configured to validate DNSSEC records can then determine if it was the correct answer. (i.e. if the answer was secured)

Here is the process in more detail:

How does DNSSEC work exactly for a resolver?

1. Resolver asks for the AAAA record for www.isc.org. Since it's configured to validate responses, it sets the "DO" ("DNSSEC OK") bit in the query which requires EDNS0 (more about the possible side effects of this later)

2. The resolver then gets a answer and tries to validate it by querying the root server for the relevant DS and DNSKEY records and validating it via a out-of-band "trust anchor".

3. Then the resolver would follow the normal DNS path checking for DS keys to verify the DNSKEY for "org" and then "isc.org".

4. At this point if it was all successful, the resolver would then verify the RRSIG record it received for www.isc.org.

5. If there is no RRSIG record for www.isc.org (or there are missing DS or DNSKEY records in the chain) then validation fails and you get a insecure response. (status set to Insecure, Bogus & Indeterminate)

Some pitfalls along the way...

EDNS and EDNS0

With the additional data required for DNSSEC, responses were going to be larger than 512 bytes which is the traditional size of a packet.

```
% dig +short +dnssec www.isc.org AAAA @sfba.sns-pb.isc.org

2001:4f8:0:2::d

AAAA 5 3 600 20100409074748 20100310074748 2658 isc.org.
TUYn3MwAqApLmZ8BTl5b1gaXvhmLxWf0KdTl3nz3+aESfLLAOyL1jaUW
HauUnshOmeYCRGIZ08OVT5EerSW+B422ItLdqPyKP1IA2RvsoSyykQKa
MwHqD9LXFSsMUCgt7Hwr7nBrshtMkeUnUVDuEZ9VAxGf4IgNn/ZJoaKr RBU=

[...]

;; MSG SIZE  rcvd: 1245
```

EDNS (**E**xtension Mechanisms for **DNS**) was developed by the IETF to allow larger sized UDP packets (from 512-4k) to carry this additional information, and EDNS0 (RFC 2671) was designated for DNSSEC.

Some pitfalls along the way...
EDNS and EDNS0

Problem -

- Device support (Home gateways, PIX, etc.)
- “Misfit toys” (“Deploy and forget”)
- Firewalls

Without the wide deployment of EDNS, resolvers will fall back to using TCP, which could cause resource issues on the upstream authoritative servers. This delayed deployment as fixes were deployed...

You can check to see if you are behind such a firewall/device by using the DNS-OARC reply-size tester @ https://www.dns-oarc.net/oarc/services/replysizetest

Some pitfalls along the way...

EDNS and EDNS0

Here are my results from the Residents network (Rogers DSL):

```
% dig +short rs.dns-oarc.net txt
rst.x476.rs.dns-oarc.net.
rst.x485.x476.rs.dns-oarc.net.
rst.x490.x485.x476.rs.dns-oarc.net.
"64.71.246.225 DNS reply size limit is at least 490"
"64.71.246.225 lacks EDNS, defaults to 512"
"Tested at 2010-05-11 12:08:11 UTC"
```

Here are my results from the guOttawa network:

```
% dig +short rs.dns-oarc.net txt
rst.x3827.rs.dns-oarc.net.
rst.x3837.x3827.rs.dns-oarc.net.
rst.x3843.x3837.x3827.rs.dns-oarc.net.
"Tested at 2010-05-11 12:58:55 UTC"
"137.122.6.9 sent EDNS buffer size 4096"
"137.122.6.9 DNS reply size limit is at least 3843"
```

Light at the end of the tunnel?

IANA (managed by ICANN) has been signing the root zone on a experimental basis since 2001.

In the fall of 2009, ICANN, Verisign & the US Government agreed to sign the root zone which was signed by the end of 2009.

The signed root zone is currently rolled out across all the root servers in a incremental process; the schedule is laid out at www.root-dnssec.org

Currently the signed root is unvalidatable.

If all goes as planned, a validatable root zone will be published on 1st July 2010.

Deployment

- **DNS Software (open-source)**
  - BIND
    - v9.3 was the first to support DNSSEC-bis
    - v9.6 is the first to support NSEC3
  - NSD
  - Unbound
  - PowerDNS
- **TLD/RIR Deployments**
  - Deployed: .bg, .br, .ch, .cz, .gov, .li, .org, .pr, .pt, .museum, .na, .se, .th, .tm, .uk
  - Testing: .at, .de, .fr, .jp, .mx, .nl, .ru, .tw
  - .com & .net coming in 2011
  - APNIC, ARIN & RIPE (in-addr.arpa)

Deployment (cont.)

ISC DLV (DNSSEC Look-aside Validation) Registry

- is an extension to the DNSSECbis protocol.
- it provides an additional entry point (besides the root zone) from which to obtain DNSSEC validation information.
- It was developed to help spur DNSSEC deployment.
- More information can be found at https://dlv.isc.org/

Web Browser Support

- Firefox plugin @ https://labs.nic.cz/dnssec-validator/

And for the OS vendors...

In FreeBSD:

How can *we* pass along a validated response in a DNSSEC-aware world? *gethostbyname-ng*? A new nsswitch provider?

Note that BIND 9.7 comes with a updated libdns library with has DNSSEC-aware getaddrinfo() and getnameinfo().

Other options?

Next Steps...

Sign your zones!

A good tutorial can be found here:

http://www.nlnetlabs.nl/publications/dnssec_howto/dnssec_howto.pdf

Turn on DNSSEC validation on your resolvers (use ISC’s DLV Registry to validate while you wait for the root to be signed)

Work to make various base libraries and utilities DNSSEC aware.