DNSSEC Deployment Challenges

Geoff Huston
APNIC
June 2016
Turning Validation on

Bind Config

```plaintext
// BIND named.conf file for RFC5011 style keyroll testing.
//
// NOTE:
// This is an example named.conf file to test RFC5011 style key rollovers.
// It is NOT useful for general purposes.
//
options {
    directory "/var/named";
    pid-file "/var/run/named/named-alt.pid";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_memstats";
    zone-statistics yes;
    // We need to allow recursion so that we can actually query the root.
    recursion yes;
    // Not much point without doing DNSSEC :-P
dnssec-enable yes;
    dnssec-validation yes;  # enable DNSSEC validation
    auth-nxdomain no;        # conform to RFC1035
    listen-on { 127.0.0.2; }; 
};
```
Why Not?

• It’s too hard
• It will take more time to resolve a name
• It will block out names with invalid DNSSEC signatures
• Too few names are signed to make a difference
• Attacks on the DNS are too rare to raise concerns
• Many folk rely on lies in the DNS
  DNS64, national content blocking measures, forced proxy redirection
• No browser wants to commit to DANE to take a positive step in cleaning up the putrid rotting security fiasco that is CA certificates today!
But maybe there is a point here

• Is having resolvers validate what they provide back to the query agent enough to improve the security of the DNS?
  – If you can intrude in an open conversation between the client and their resolver then MITM attacks in the DNS can still take place
How can we improve this situation?

Some potential directions

– Push validation back to the client application
  • Such as GetDNS

– Secure the conversation between the application and a trusted resolver
  • Such as https://dns.google.com

– (re)introduce DANE to browsers using DNSSEC credential stapling
  • https://www.imperialviolet.org/2011/06/16/dnssecchrome.html
Thanks!

**DNSSEC Reports:**  http://stats.labs.apnic.net/dnssec
Additional Slides: Why Not?

- It’s too hard
- It will take more time to resolve a name
- It will block out names with invalid DNSSEC signatures
- Too few names are signed to make a difference
- Attacks on the DNS are too rare to raise concerns
- Many folk rely on lies in the DNS
- DNS64, national content blocking measures, forced proxy redirection
- No browser wants to commit to DANE to take a positive step in cleaning up the putrid rotting security fiasco that is CA certificates today!
Why Not?

• It’s too hard
  • It will take more time to resolve a name
  • It will block out names with invalid DNSSEC signatures
  • Too few names are signed to make a difference
  • Attacks on the DNS are too rare to raise concerns
  • Many folk rely on lies in the DNS
    DNS64, national content blocking measures, forced proxy redirection
  • No browser wants to commit to DANE to take a positive step in cleaning up the putrid rotting security fiasco that is CA certificates today!

One line of config in a recursive resolver!
Why Not?

• It’s too hard
• It will take more time to resolve a name
• It will block out names with invalid DNSSEC signatures
• Too few names are signed to make a difference
• Attacks on the DNS are too rare to raise concerns
• Many folk rely on lies in the DNS

Cached answers will take no longer to resolve from a validating resolver as compared to a non-validating resolver

Retrieving DNSSEC credentials take queries, and queries take time

Currently, DNSSEC validation queries are serialized in most resolvers. This time could be reduced if these queries were parallelised.

As with all things in the DNS, this is not necessarily true.
Why Not?

• It’s too hard
• It will take more time to resolve a name
• It will block out names with invalid DNSSEC signatures
• Too few names are signed to make a difference
• Attacks on the DNS are too rare to raise concerns
• Many folk rely on lies in the DNS
  DNS64, national content blocking measures, forced proxy redirection
• No browser wants to commit to DANE to take a positive step in cleaning up the putrid rotting security fiasco that is CA certificates today!

Yes, that’s what it’s meant to do!
Why Not?

• It’s too hard
• It will take more time to resolve a name
• It will block out names with invalid DNSSEC signatures
• Too few names are signed to make a difference
• Attacks on the DNS are too rare to raise concerns
  
But DNSSEC has incremental outcomes:

You can try and improve the integrity of YOUR name by signing it with DNSSEC
Why Not?

- It’s too hard
- It will take more time to resolve a name
- It will block out names with invalid DNSSEC signatures
- Too few names are signed to make a difference
- Attacks on the DNS are too rare to raise concerns

That assumes structural DNS censorship is not in and of itself an attack on the integrity of the DNS.
Why Not?

- It’s too hard
- It will take more time to resolve a name
- It will block out names with invalid DNSSEC signatures
- Too few names are signed to make a difference
- Attacks on the DNS are too rare to raise concerns
- Many folk rely on lies in the DNS
  DNS64, national content blocking measures, forced proxy redirection

True — but what do users want from the DNS!
Why Not?

• It’s too hard
• It will take more time to resolve a name
• It will block out names with invalid DNSSEC signatures
• Too few names are signed to make a difference
• Attacks on the DNS are too rare to raise concerns
• Many folk rely on lies in the DNS

  DNS64, national content blocking measures, forced proxy redirection

• No browser wants to commit to DANE to take a positive step in cleaning up the putrid rotting security fiasco that is CA certificates today!

  is ever so slightly faster really better than vulnerability to third party attack via compromised CAs?