Some DNSSEC thoughts

Geoff Huston
Chief Scientist, APNIC
June 2007

The DNS is a miracle!

You send out a question into the net ...

And an answer comes back!

Somehow

- But ...
  - WHO provided the answer?
  - Is it a REAL answer?
  - Can I TRUST the answer?

DNSSEC – The Motivation

- How can a DNS resolver tell if a DNS response can be trusted as **authentic**?
- Is this the **correct** DNS response?
  - Has it been altered?
  - Has it been truncated?
  - Is it hopelessly out of date?

DNSSEC – The Theory

Sign and publish ***everything***!

- Every DNS zone has associated key pairs
- Each zone publishes:
  - The public key (DNSKEY RR)
  - Private-key signatures of all RR Sets (RRSIG RR)
  - Private-key signed “gaps” in the zone file (NSEC RR)
  - Hashes of the public key of child zones (DS RR)

So you take a small zone....

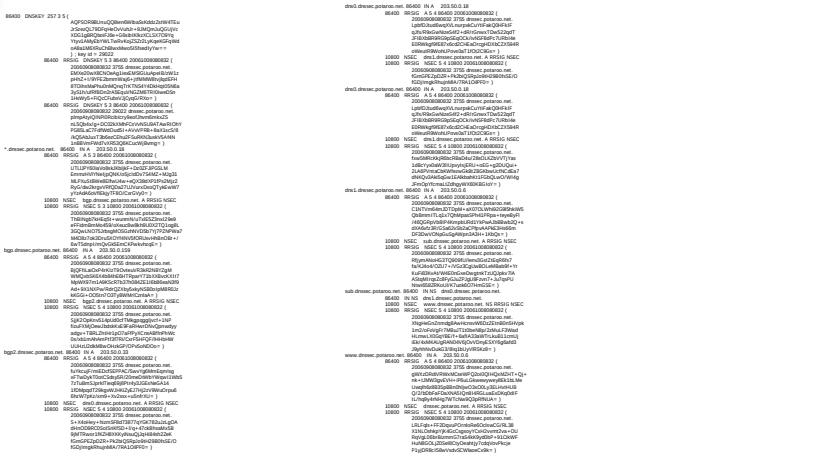
```
TTL 86400
$ORIGIN         dnssec.potaroo.net.
@               IN      SOA     dns0.potaroo.net. gih.potaroo.net. (2006090803 3h  15   1w   3h )

; name servers
                IN      NS      dns0.potaroo.net.
                IN      NS      dns1.potaroo.net.
;
; subdomains
;
sub             IN      NS      dns0.dnssec.potaroo.net.
                IN      NS      dns1.dnssec.potaroo.net.
;
www             IN      A       203.50.0.6
bgp             IN      A       203.50.0.159
bgp2            IN      A       203.50.0.33
dns0            IN      A       203.50.0.18
dns1            IN      A       203.50.0.6
;
; wildcard
;
*               IN      A       203.50.0.18
```

And turn it into a big zone...

```
dnssec.potaroo.net.     86400   IN SOA  dns0.potaroo.net. gih.potaroo.net. (
                                2006090803 ; serial
                                10800      ; refresh (3 hours)
                                15         ; retry (15 seconds)
                                604800     ; expire (1 week)
                                10800      ; minimum (3 hours)
                                )
                        86400   RRSIG   SOA 5 3 86400 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                syLogFkxP1KtEkYp4Pic6ggW1Nr16powIzx+
                                VbpdA/erzxRdARd1I77F56N7TB+v3aS82aLh
                                BLIN+f0MzHEo/JNWVl0xjn95pRDd3gyZSoE+
                                aWG21MokMbTBxF2pYmFA1ENNKKK+pSXuXvsS
                                dAP+kcVqT6PfO67+m2chsqbh+uA= )
                        86400   NS      dns0.potaroo.net.
                        86400   NS      dns1.potaroo.net.
                        86400   RRSIG   NS 5 3 86400 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                p2kKLK4gzlm8nkr4lpXyz4FirWWXtiyXc5X/
                                Ns2NYC3CNYDNIRFHzEI14RZO08R9z4aoQlfO
                                jXidiJZ2BgxzmykVJUaA7AwGirVtr+6wDJrd
                                if9tm7UdYN2powrP9o2tq0DKhwYk8i4Dyjdd
                                9kwt7/x44ZEcEj7w30GfW4uvy8= )
                        10800   NSEC    *.dnssec.potaroo.net. NS SOA RRSIG NSEC DNSKEY
                        10800   RRSIG   NSEC 5 3 10800 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                h75DS6C1IGLPRbqtz9+KV4oSuidA+Bdt6geq
                                q6NRrneNGA6Rr00FK4Td9AQS1+JpM3KriDl5
                                LKqQM7yMarC7aE3v/23iW9YqFv3Z6PpjW7Ze
                                oEhaLNCV3kG4tVmILsoGEp/EWtgNTnXkJdkD
                                hW+o91s7XVnGmO7m9JkUOu8sS2E= )
                        86400   DNSKEY  256 3 5 (
                                AQO8xvbN4hZ8bn926wpM8c9Uqqhqcf45v73k
                                4JYSu+6o/QsPCKwJoDYxMH3s5Z0NJlgLUQs
                                ciZZKDYVHPW3Txt59bHrn739osnQ80RbOQVT
                                H/Vi//L3BGjZrZr+PWtH2Vb3wIhrujMej2m4
                                E2Mth/XjSDAhYZVWCNhJG0nPH6G6Ww==
                                ) ; key id = 3755
```

There's more ...!



DNSSEC – Signing a Zone

- Generate a keypair
- Generate a Key-Signing keypair
- Load the keys into the zone
- Use a zone signing utility to sign **every** RR in the zone, and to sign **every** name gap in the zone
- Update the parent zone with the child's public key hash
- Publish the zone with a DNSSEC-aware name server

DNSSEC – DNS Response

- The *Additional Information* section in a DNSSEC response contains:
  - a **DNSKEY RR**, and
  - an **RRSIG RR** for a data response, or
  - an **NSEC(3) RR** response for a "no such data" response

DNSSEC – Response Validation

- Validation of a DNS response:
  - Did the matching private key sign the RRSIG RR?
  - Does the hash match the RR data?
  - Does the public key validate?
    - Does the parent have a DS RR?
    - Has the Parent signed the matching RRSIG RR?
    - Does the parent's key validate?
    - Loop until you get to a recognised "trust anchor"

This interlocking of parent signing over child is a critical aspect of the robustness of DNSSEC. It's also DNSSEC's major weakness today!

Some Questions:

- How do you know if this is current data or a replay of older stale data that was signed with the current key?
- How do you know that a zone is DNSSEC signed?
- How do you roll keys over?
- How do you revoke keys?
- What's NSEC3?
- What's a "trust anchor"?

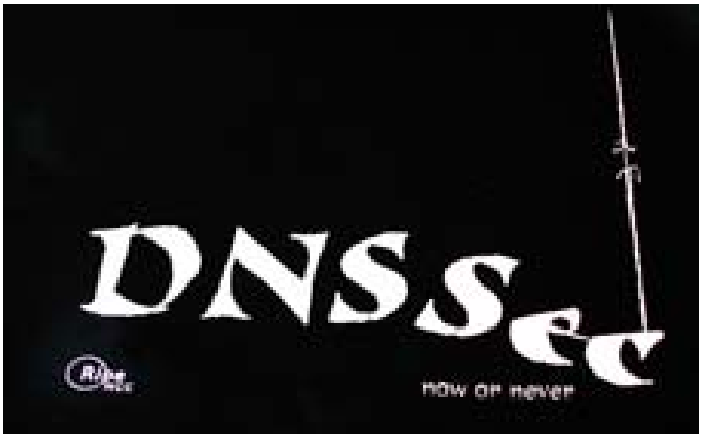
"Trust" is a very tricky thing

- In the ideal world ALL the DNS would be DNSSEC signed
  - As long as you have the current root DNSSEC public key as your trust anchor then every DNS response can be validated by simply walking backwards up the name hierarchy to the root
- But this is really not the case:
  - Only a few zones are signed
  - And you don't know which!
  - So which trust keys do you load and from whom?
  - And when should you update these keys?
  - Right now DNSSEC is unuseable!

Status of DNSSEC

- The DNSSEC spec is over 10 years old
- Interest in deployment of DNSSEC has been very limited
- Will DNSSEC ever be deployed?

One Opinion



DNSSEC Positives

- DNSSEC makes the DNS harder to attack
- Trust injection into the DNS can be more than just trusting the DNS
  - Use the DNS to pass other keys, other data objects, secured by DNSSEC
- DNSSEC can avoid the overheads of yet more special-purpose PKIs

The DNS is a critical point of vulnerability in the network's overall model of integrity of operation -- DNSSEC can help here

DNSSEC Negatives

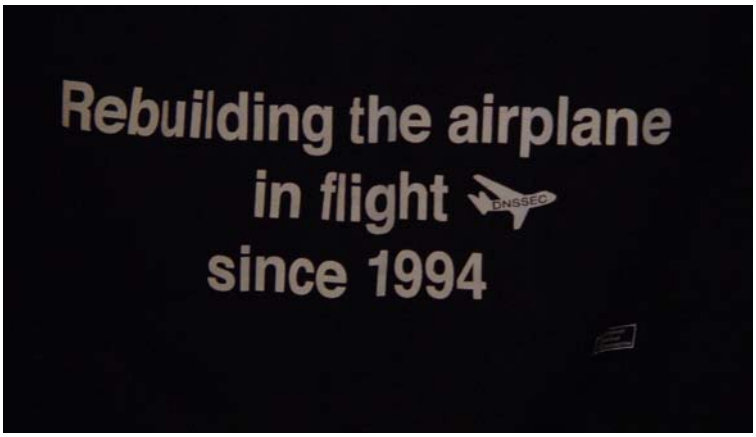
- DNS Zones get VERY LARGE
  - x 10 in size
- DNS responses can get VERY LARGE
- DNSSEC Zone management is complicated
- NSEC implicitly exposes the zone contents
- NSEC3 is extremely obscure and challenging to verify
- Who can use the signed answer, and how?
- The partial deployment trust model is useless

DNSSEC represents a significant investment on the part of the server with unclear benefits for the client

Next Steps for DNSSEC?

- Complete, top down, all zones, DNSSEC deployment looks like it may never happen
- If all that happens is that only some of us deploy DNSSEC, then the entire DNS effort is largely a waste of time, because of the trust point discovery problem in the current DNSSEC model
- Can we devise a more robust partial deployment model that can deliver benefits to both the DNSSEC signed zone publisher and the DNSSEC-aware resolver client base?
  - Is the DLV model of interest here?
  - Are there other approaches?

Another Opinion



My Opinion

- The DNS would be really very useful if everyone deployed DNSSEC
- The DNS would be far more cumbersome, far more complex to manage, and far more error-prone to operate, if everyone deployed DNSSEC
- And while only some of us deploy DNSSEC its not of much value at the moment!

Thank You