Defending your DNS in a post-Kaminsky world

Paul Wouters
<paul@xelerance.com>
DNS resilient

Longer TTL's are much safer

The calculations above indicate the relative ease with which DNS data can be spoofed. For example, using the formula derived earlier on a domain with a 3600 second TTL, an attacker sending 7000 fake response packets/s (a rate of 4.5M/s), stands a 10% chance of spoofing a record in the first 24 hours, which rises to 50% after a week.

For a domain with a TTL of 60 seconds, the 10% level is hit after 24 minutes, 50% after less than 3 hours, following 90% after around 9 hours.

Note that the attacks mentioned above can be thwarted by watchful server operators -- an unexpected incoming stream of 4.5M/s of DNS packets might be noticed.

An important assumption in these calculations is a known or constant static destination port of the authentic response.

The IETF

DNSSEC must happen NOW

Sunday, July 14, 2008

Cryp.to News

Sunday, August 17, 2008

The case against DNSSEC

I was talking to my good friend Venet Enwhoate the other day when he suddenly turned to me and said "I don't think we need DNSSEC". Sharp intake of breath. Transcribed after a long and involved discussion of his case boiled down to four points:

1. SSL provides known and trusted security. DNSSEC follows the same flawed model; it is superfluous.
2. DNSSEC is complicated and potentially prone to errors. As The Onion makes it clear, DoS attacks would be trivial.
3. DNSSEC does not solve the last mile problem.
4. DNSSEC is complex, but the solution does not change.

CircleID

DNSCurve will save the day

Bernstein said that DNSSEC offers "a surprisingly low level of security" while causing severe problems for DNS time on breakable Ren patches." Bernstein said follow.

He called for development imp of DNSSEC alternatives that quickly and securely. The
Vendor and NGO's involved
Two phase deployment

First release a generic fix for the Kaminsky attack that does not leak information to the bad guys (source port randomization)

Then release the bug and patches specifically against the Kaminsky attack
DNS query packet

- IP header containing Source IP and Dest IP
- UDP or TCP Header containing Source Port and Dest Port (if TCP, also random Sequence Number)
- DNS Query ID
- DNS Query
- Option flags
DNS query example

12.110.110.204 → 193.110.157.136

UDP:12345 → 53

DNS Query ID: 54321
DNS Question: www.ripe.net?
Option flags: RD
DNS Answer packet

193.110.157.136 → 12.110.110.204

UDP:53 → 12345

QUESTION SECTION
Query ID: 54321
Question: www.ripe.net?

ANSWER SECTION
www.ripe.net = 193.0.0.195 (ttl=172800)

AUTHORITY SECTION
ripe.net NS ns-pri.ripe.net. (ttl=172800)
ripe.net NS ns-ext.isc.org. (ttl=172800)

ADDITIONAL SECTION
ns-pri.ripe.net A 193.0.0.195 (ttl=...)
ns-pri.ripe.net AAAA 2001:610:240:0:53:3
TXID is not enough anymore

Bellowin's (theoretical) attack (1995)
Losing the race

---

**EndUser**

ISP NS

**EVIL**

RBC NS

---

**TIME**

---

**Q:** www.rbc.com?

---

**Q:** www.rbc.com? TXID = 32768

---

**A:** www.rbc.com = 1.2.3.4; TXID = 00001

---

**A:** www.rbc.com = 1.2.3.4; TXID = 00002

---

**A:** www.rbc.com = 1.2.3.4; TXID = 00003

---

**A:** www.rbc.com = 142.254.1.143; TXID = 32768

---

**TTL=86400**
Winning the race
Random source ports

Bernstein: Use random src ports as entropy
DJB's hack is still just a hack
Birthday Attack on src ports
NAT and DNS rebinding

EndUser ➔ Nameserver ➔ NAT / Firewall

EndUser: Q:www.rbc.com
TXID:32768
SRC PORT:54195

Nameserver: Q:www.rbc.com
TXID:32768
SRC PORT:1025

RBC NAMESERVER ➔ EVIL

Black Hat Briefings
NAT and DNS rebinding (2)
Kasphureff's attack (1997) caused Bailywick restrictions.
What protected our DNS?

- The attacker cannot see your packet
  You always lose at StarBucks and TOR
- Transaction ID (TXID)
- Time To Live (TTL)
- Bailywick
Without source port randomization, this only takes about 65535 packets
DNS related issues: Double Fast Flux

- Botnets use domains with NS and A records with low (eg 3 minute) TTL's.
- Change NS records via Registrar very quickly too (hours).

This makes them next to impossible to shutdown.

(and soon OpenDNS commercial double fast flux)
DNS related issues:

The Wifi hotspot

- Captive portals using DNS with mini DNS “server”
- This is so they can serve fake DNS
- This can cause client to cache wrong DNS
- Bad implementations break on EDNS and DNSSEC (hardcoded bits checking)

Use transparent IP proxy instead
Where to fix the DNS?
DNS is critical infrastructure

- Backwards compatible (opt-in)
- Non-invasive or intrusive (drop-in)
- Non-disruptive (no CPU/Bandwidth hog)
- No Protocol changes (we have DNSSEC)
- Preferably no TYPE overloading
- No magic such as untested cryptography
- Patent / Royalty free
Thou Shalt Implement:

**BCP 38**

(Egress Filtering)
Thou Shalt not:

combine a recursive and authoritative server
Authoritative nameservers

Upgrade server to allow DNSSEC
Diversify your infrastructure

```sh
dig 9.6.0a1 ns xelerance.com
```

```bash
;; QUESTION SECTION:
xelerance.com.       IN  NS

;; ANSWER SECTION:
xelerance.com.       844 IN  NS  ns2.xelerance.org.
xelerance.com.       844 IN  NS  ns0.xelerance.nl.
xelerance.com.       844 IN  NS  ns1.xelerance.net.

;; ADDITIONAL SECTION:
ns0.xelerance.nl.     972 IN  A   193.110.157.135
ns1.xelerance.net.    96036 IN  A   209.237.247.134

;; Query time: 118 msec
;; SERVER: 193.110.157.2#53(193.110.157.2)
;; WHEN:  Sat Jan 31 12:05:29 2009
;; MSG SIZE  rcvd: 142
```
Network IDS / Firewall

- It's patchwork (pun intended)
- Does not address the problems
- Cannot make a decision when an attack is detected. What to do? Blocking is bad (denial of service to yourself)

Monitor, log and warn. Do not interfere
Be very careful with DNS load balancers
Monitor Unix based DNS

Unbound DNS answers by return code - by week

<table>
<thead>
<tr>
<th>Code</th>
<th>Current</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDERROR</td>
<td>639.17m</td>
<td>224.19m</td>
<td>1.50</td>
<td>24.27</td>
</tr>
<tr>
<td>SERVFAIL</td>
<td>107.65m</td>
<td>21.70m</td>
<td>219.53m</td>
<td>6.22</td>
</tr>
<tr>
<td>NXDOMAIN</td>
<td>527.49m</td>
<td>223.40m</td>
<td>1.09</td>
<td>13.09</td>
</tr>
<tr>
<td>nodata</td>
<td>76.60m</td>
<td>7.75m</td>
<td>184.85m</td>
<td>3.37</td>
</tr>
<tr>
<td>answer secure</td>
<td>25.50m</td>
<td>0.00</td>
<td>6.86m</td>
<td>776.15m</td>
</tr>
<tr>
<td>answer bogus</td>
<td>71.15m</td>
<td>17.68m</td>
<td>159.57m</td>
<td>6.18</td>
</tr>
<tr>
<td>num rrsets marked bogus</td>
<td>18.87m</td>
<td>0.00</td>
<td>26.87m</td>
<td>445.58m</td>
</tr>
</tbody>
</table>

Last update: Sun Feb 1 11:10:03 2009
Monitoring using Cisco

www.cisco.com/web/about/security/intelligence/dns-bcp.htm

policy-map type inspect dns preset_dns_map
parameters
  !--- TXID matching – allow only 1 response
dns-guard
id-randomization
id-mismatch count 10 duration 2 action log
message-length maximum 512
match header-flag RD
drop
Monitoring using Cisco

```bash
firewall# show service-policy inspect dns
```

**Global policy:**

Service-policy: global_policy

Class-map: inspection_default

Inspect: dns preset_dns_map, packet 37841, drop 0, reset-drop 0

message-length maximum 512, drop 0
dns-guard, count 21691
protocol-enforcement, drop 0
nat-rewrite, count 0
id-randomization, count 21856
id-mismatch count 10 duration 2, log 2
Application fixes

So many different applications to fix
DNS API for applications is poor
Easy to fool: DNS Rebinding or Fast Flux
But let's not build DNS recursive nameservers in every application

(however a good recursive dns server on each host is a good solution)
The inevitable:

Fix recursive nameservers

Port randomization
Sanitize TTL's
Use more IP addresses per DNS server
Harden against bogus size packets
Harden glue
Additional queries for infrastructure data
0x20
Birthday Attack protection

- Do not allow multiple queries for the same question to be outstanding (AKA query chaining)

  Unbound, Bind and PowerDNS implement this properly

dnscache from DJB was apparently vulnerable to this until a few days ago!
Rebinding protection

Allow to specify IP addresses that may never appear in “external” domain names

This way you can ensure 10.1.1.0/24 would never come in through DNS rebinding.
(supported in Unbound and PowerDNS)
Attacks can be detected
Attack response #1

- At a spoof detection threshold, ignore all answers for that query
- Prevents accepting the right forged answer
- Also prevents accepting the real answer

spoofmax=?

Small value : easy DOS
Large value: might be too late
(PowerDNS has spoofmax=20)
Attack response #2

- At a spoof detection threshold throw away the **entire** cache and start from scratch
  - Prevents using an accepted forged answer
  - Small value: easy DOS on the cache
  - Large value: might be too late
    - (Unbound has spoofmax=10M)
Chain your caches (esp. the ones behind NAT)
Add more NS records?

- If you already have at least two or three, this does not buy you much
  Only makes an attack marginally harder

- Excessive NS records cause other problems (and adds more potentially outdated / vulnerable nameservers)
Pick nameserver more random

- Old days: prefer nameserver with shortest TTL
- New ways: Add some fuzz
Hardening infrastructure queries

Before accepting NS records or A records of nameservers, ask at least two different nameservers.

Before accepting glue records or additional data, independently verify these with new queries.

(extra work is only needed once, then we use caching – minimum impact)
The 0x20 defense (Paul Vixie)

DNS Question: bogus12345.www.paypal.com?
Option flags: RD
The 0x20 defense (Paul Vixie)

DNS Question: bogus12345.www.paypal.com?
Option flags: RD

DNS Query ID: 54321
DNS Question: bOGus12345.Www.pAYpaL.Com
The 0x20 defense (Paul Vixie)

DNS Query ID: 54321
DNS Question: BoGUs12345.Www.pAYpaL.com

QUESTION SECTION
Query ID: 54321
Question: BoGUs12345.wWW.pAYPa.l.cOM

ANSWER SECTION

AUTHORITY SECTION
bogus12345.www.paypal.com NS
www.paypal.com

ADDITIONAL SECTION
www.paypal.com A 1.2.3.4
The 0x20 defense (Paul Vixie)

- You don't need “Td-CaNAdaTRuSt.cOm” when you can get “.CoM”

- Fails completely for the root (“.”)
Double Fast Flux protection

Draft-bambenek-doubleflux suggests:

- Replacing the TTL's of NS and A records of NS records with TTL=72 hours.
- Limit Registrar changes to once per 72h.
- Recursors and clients should drop NS or A of NS with TTL < 12.
The inevitable:
Fix recursive nameservers

RFC 5452 “Measures for Making DNS More Resilient against Forged Answers”
draft-wijngaards-dnsext-resolver-side-mitigation
draft-vixie-dnsext-0x20
The real solution

DNSSEC
What is DNSSEC?

- Authenticate (non)existence of data within a zone
- Create a path of trust between zones
- Sign and preload the root ("." key)
Traditional DNS

The root ("."), .com, xelerance.com
Add a public key to zone
Sign zone with private key
Give hash(pubkey) to parent
Rinse and Repeat
New DNS Record types

- **DNSKEY**: Public key
- **RRSIG**: Signature RRset
- **NSEC**: “Clever” Record denial of existence
- **NSEC3**: “Super Clever” Record stealthy denial of existence
- **DS**: Delegation Signer r.
DNSSEC answers can be:

- **SECURE**  
  Validated with key

- **INSECURE**  
  Validated but no key

- **BOGUS**  
  validation failed

- **UNKNOWN**  
  ServFail etc
DNSSEC bits

- The DO bit (query)  DNSSEC (is) OK
- The AD bit (answer)  Authenticated Data
- The CD bit (query)  Checking Disabled
New DNSSEC errors

- Uhm, none. For maximum compatibility. If any error happens, return the old ServFail.

- A validator can then redo the query with the CD bit if it wants to see why it failed.
Let's see some DNSSEC...

Unlike Adam Laurie and Johnny Long, I have no cool Hollywood clip I can show.
<table>
<thead>
<tr>
<th>Domain</th>
<th>TTL</th>
<th>Type</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>bofh.xelerance.com.</td>
<td>3600</td>
<td>IN A</td>
<td>193.110.157.17</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>RRSIG</td>
<td>A 5 3 3600 20090314165933</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20090212165933 16352 xelerance.com.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ohgc1aigYWldUYr13xQRjCNdleLtaQC1sXp[...])</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>NSEC</td>
<td>bugs.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20090212165933 16352 xelerance.com.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H5Cr4Z8ovj81fwCCHBvOi2fiD3zX25ND4At[...])</td>
</tr>
<tr>
<td>bugs.xelerance.com.</td>
<td>3600</td>
<td>IN A</td>
<td>193.110.157.129</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>RRSIG</td>
<td>A 5 3 3600 20090314165933</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20090212165933 16352 xelerance.com.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dmWVWxKQYXqzxWwCNwH3jdGTWqwQE5PHFPR[...])</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>NSEC</td>
<td>build.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20090212165933 16352 xelerance.com.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NLTif8GabVXmtnWKUtiAGkHD5dPr+yGhAgM[...])</td>
</tr>
<tr>
<td>build.xelerance.com.</td>
<td>3600</td>
<td>IN A</td>
<td>193.110.157.194</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>RRSIG</td>
<td>A 5 3 3600 20090314165933</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20090212165933 16352 xelerance.com.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nE9pOj6e2aAT+B76jH0dMqIKy6+PwIibB4s[...])</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>NSEC</td>
<td>calendar.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td></td>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20090212165933 16352 xelerance.com.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lfk6E0dqyubGeDqi7z75004x3mtFNpPpg0wTr[...])</td>
</tr>
<tr>
<td>calendar.xelerance.com.</td>
<td>3600</td>
<td>IN A</td>
<td>193.110.157.130</td>
</tr>
</tbody>
</table>
DNSSEC in a nutshell

;<<>> DiG 9.6.0a1 <<>> +multiline +dnssec -t ds nic.cz @193.110.157.136
; ; '-HEADER]-' opcode: QUERY, status: NOERROR, id: 44991
; ; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 7, ADDITIONAL: 1
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags: do; udp: 4096
; ; QUESTION SECTION:
; ;nic.cz.
; ; ANSWER SECTION:
;nic.cz. 445 IN DS 59916 5 1 (144130216E45C4EC2BB8595E817916E8B060D87B)
nic.cz. 445 IN DS 27979 5 1 (FF11E740A0254EC63C738A47E52ABF3AD91D8C43)
nic.cz. 445 IN RRSIG DS 5 2 1800 20090314003628 (20090212003628 4092 cz.
c4p82mdTbbydViihi9HP8f8k1qNOnWYfJemdAF7Zk78L/[...])
; ; AUTHORITY SECTION:
cz. 16645 IN NS d.ns.nic.cz.
cz. 16645 IN NS f.ns.nic.cz.
cz. 16645 IN NS a.ns.nic.cz.
cz. 16645 IN NS c.ns.nic.cz.
cz. 16645 IN NS e.ns.nic.cz.
cz. 16645 IN NS b.ns.nic.cz.
cz. 16645 IN RRSIG NS 5 1 18000 20090313023545 (20090211023545 4092 cz.
xONjUdAH7ieDwrVK3En/CmV0oM6JJUTiF5QczRuscHrM[...]
NSEC: Denial of existence

<table>
<thead>
<tr>
<th>3600</th>
<th>NSEC</th>
<th>_sip._tcp.xelerance.com. A NS SOA MX TXT NAPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 2 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>_sip._udp.xelerance.com. SRV RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 4 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>admin.xelerance.com. SRV RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 4 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>aivd.xelerance.com. A SSHFP RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>conference.aivd.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>monitor.ams.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 4 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>bofh.xelerance.com. CNAME RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 4 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>bugs.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>build.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>calendar.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>calender.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
<tr>
<td>3600</td>
<td>NSEC</td>
<td>cdc.xelerance.com. A RRSIG NSEC</td>
</tr>
<tr>
<td>3600</td>
<td>RRSIG</td>
<td>NSEC 5 3 3600 20090314165933 (</td>
</tr>
</tbody>
</table>
NSEC3: denial of existence with a hack

- Do not use names, but hashes
- For added work, hash X times
- Now sort the hashes

The validator that gets an NSEC3 record back, hashes the QUERY name (x times) too and compares
; <<>> DiG 9.6.0a1 <<>> +multiline +dnssec -t ns hhhh.gov @193.110.157.136
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 1

;; AUTHORITY SECTION:
gov. 86381 IN SOA A.GOV.ZONEEDIT.COM. govcontact.ZONEEDIT.COM. (1234994462 ; serial
3600 ; refresh (1 hour)
900 ; retry (15 minutes)
1814400 ; expire (3 weeks)
86400 ; minimum (1 day)
)
gov. 86381 IN RRSIG SOA 7 1 259200 20090223210103 (20090218210103 31802 gov.
kF4kRKyTIok/tuMdrBB+fsmm5+9HYunPGu05292z3+B1[...])

VVSOMCNUB7A79EALVJEH4VN12192C715.gov. 86381 IN NSEC3 1 0 10 ABAB 0002H1U5Q5HGQCIDMSB0QRETCK0N6FLT NS S0A RRSIG DNSKEY NSEC3PARAM
VVSOMCNUB7A79EALVJEH4VN12192C715.gov. 86381 IN RRSIG NSEC3 7 2 86400 20090223210103 (20090218210103 31802 gov.
SazLR\NSeO39Cn0fzWDs/zI8g4qFw5Mm61vZ9neuptfG[...])
g0YCA6nrzJDKAkwNlTXLLnfA6kOvyJdfA==

AJBACCGUPENCE2AA1RNHHLUFHA37G18F.gov. 86381 IN NSEC3 1 0 10 ABAB AJFCCN9I570TBLMTTFS3H3IREPV0I9TJ NS
AJBACCGUPENCE2AA1RNHHLUFHA37G18F.gov. 86381 IN RRSIG NSEC3 7 2 86400 20090223210103 (20090218210103 31802 gov.
0KfqMdW4sV9tvFVH/FY45EPYa53C1qD2px37m2J5a9h8
DNSSEC: Use Zone and Key Signing keys
DNSSEC: Key Signing Key Rollover
DNSSEC: Key update
Triggers or Timers?

For DNSSEC: Key update from child to parent
For most domains: Any updates via Registrant to Registrar to Registry
For some domains: Registrant – Registry communication
Most common solution will be EPP via Registrar. Some by Registry polling
DNSSEC Look-aside Verification

1) Look for KEY for .com

2) Look for key for xelerance.com
Feb 16: https://itar.iana.org/

IANA provides an Interim Trust Anchor Repository to share the key material required to perform DNSSEC verification of signed top-level domains, in lieu of a signed DNS root zone. This is a temporary service until the DNS root zone is signed, at which time the keying material will be placed in the root zone itself, and this service will be discontinued.

What is the repository for?

The Interim Trust Anchor Repository, or ITAR, acts as a mechanism to disseminate “trust anchors” that have been provided by the operators of top-level domains who use DNSSEC to secure their zones. IANA is responsible for managing the DNS root zone, and uses these existing trust relationships to verify the supplied trust anchors come from the correct party. The system is considered interim as it is designed to be deprecated once the DNS root zone itself is signed with DNSSEC.

What is a beta?

This is a preliminary testing version of the service for the community to try. We will take feedback and improve the product before it is considered fully production ready. In particular, we appreciate feedback on problems that occur, as well as features that could be added to make the service more useful. You can send any comments to itar@iana.org.

Who may submit trust anchors?

This repository is limited to trust anchors for top-level domains. Top-level domain operators who have been issued trust anchors by ICANN may submit trust anchors to IANA through this repository.

Browse the trust anchor repository
Download the trust anchors
Master File Format
XML

How to use
Processes and Procedures
Add a trust anchor
Revoke a trust anchor
.gov is signed!

DNSSEC for All Top Level .GOV Domains

Published: August 29th, 2008 | Category: Security Vulnerabilities

Last week the Office of Management and Budget released memoranda M-08-23, titled Securing the Federal Government’s Domain Name System Infrastructure. The document states that all US government top level .gov domains will use DNSSEC starting in January 2009. This is in response to the DNS cache poisoning attack that Dan Kaminsky made public a few months ago.

New Policy

This memorandum addresses two important issues in following through with the existing policy and expanding its scope to address all USG information systems.

A. The Federal Government will deploy DNSSEC to the top level .gov domain by January 2009. The top level .gov domain includes the registrar, registry, and DNS server operations. This policy requires that the top level .gov domain will be DNSSEC signed and processes to enable secure delegated sub-domains will be developed. Signing the to level .gov domain is a critical procedure necessary for broad deployment of DNSSEC, increases the utility of DNSSEC, and simplifies lower level deployment by agencies.

B. Your agency must now develop a plan of action and milestones for the deployment of DNSSEC to all applicable information systems. Appropriate DNSSEC capabilities must be deployed and operational by December 2009. The plan should follow recommendations in NIST Special Publication 800-81 “Secure Domain Name System (DNS) Deployment Guide,” and address the particular requirements described in NIST.
The Keys to Deploying DNSSEC: Managing and Meeting Your OMB Domain Name

Thursday, March 12, 2009
Session: 8:30AM - 4:30PM
Presented by:

DNSSEC Development Coordination Initiative

The DNSSEC Deployment Initiative works to encourage all sectors to voluntarily adopt security measures that will improve security of the internet's naming infrastructure, as part of a global, cooperative effort that involves many nations and organizations in the public and private sectors.
dnssec-conf

www.xelerance.com/software/dnssec-conf

Provides key management and dnssec configuration for Fedora/RHEL/CentOS

Yum install dnssec-conf
dnssec-configure –dnssec=on –dlv=on
DNSSEC software

Authoritative nameservers:
  - Bind - www.isc.org
  - NSD - www.nlnetlabs.nl/projects/nsd/
  - Microsoft DNS (support recordtypes, not signing)

Recursive validating nameservers:
  - Bind - www.isc.org/bind/
  - Unbound - www.unbound.net
TODO: Integration

- Integrate DNSSEC resolver with Network Manager
- Use DNS caching infrastructure via DHCP obtained DNS servers, but:
- Validate all crypto ourselves on the endnode
If you have not implemented DNSSEC, are you planning to implement it?

- YES 85%
- NO 10%
- Unsure 6%
If you have not implemented DNSSEC, when are you planning to implement it?
Conclusions (1)

**Update** your nameservers, or place them behind new nameservers.

Look into more software then just Bind **Unbound**, PowerDNS recursor

Take a fresh look at your deployment, even when using firewalls and NAT. DNS *will* go through those.

Ditch DNS captive portals and broken DSL routers
Conclusions (2)

Prepare for DNSSEC

Tell your vendor[*] you require DNSSEC validation on your laptop using a DHCP obtained DNS caching server as forwarder.

[*] If you use Linux/BSD/OSX, why have you not installed/configured/enabled it yet?
Questions?

(feel free to test with nssec.xelerance.com)
Why DNSCURVE sucks

- There is no formal specification nor formal implementation, just proof of concept code
- Encrypts and protects TRANSPORT of dns data not data INTEGRITY itself
- Everyone has to bypass dns caches (or blindly trust them).
- Causes massive increase in DNS traffic
- Type overloading of NS records with long crypto keys as names (HACK)
- Uses patent encumbered Elliptic Curve cryptography
- Uses Bernstein's specifically picked homegrown elliptic curve
- No cipher or algorithm migration path if the curve falls over
- Uses 95% more CPU (on each query instead of once on a signer machine)
- Provides no partial deployment support (Secure Entry Points)
- I still need to punch him in the face for qmail