Securing the DNS, one zone at a time

Andrew Ruthven

Catalyst IT Ltd

LCA2008

Why the Domain Name System?

- Who can remember 202.78.240.49?
- How about all the other hosts you use?
- Used to use the `hosts` file.
- It started small, ended *huge* and unmanageable!

What is the Domain Name System?

- Easily Remembered Hostname → IP address mapping
  - www.catalyst.net.nz → 202.78.240.49
- Key → Value
  - 49.240.78.202.in-addr.arpa → www.catalyst.net.nz
  - 7.6.2.2.9.9.4.4.4.6.e164.arpa → sip:4992267@voip.catalyst.net.nz
- The namespace is broken into zones, at the full stops.
- Control of those zones is delegated, i.e., Catalyst IT maintains catalyst.net.nz.
- Catalyst's nameservers are **authoritative** for catalyst.net.nz.
- Globally distributed database, with replication and caching

What is right?

- It works!
- It has scaled
- Extensible
  - Extra resource record types have been added
  - Original resource record types: A, CNAME, HINFO, MX, NS, PTR, SOA, TXT, ...
  - Extras used everyday: SRV, SPF, DNAME, NAPTR, ...

This means...

- It is relied upon
- It is invisible
- But it is actually insecure

What is wrong?

- UDP 512 byte limit
- Non-existence of TLDs (i.e., `.local`)
- Using UDP, not TCP
- Network providers blocking TCP port 53
- Possible extra load due to IPv6?
- Phishing due to Unicode (IDN)
- No data integrity (beyond IP layer)
- No data origin authentication

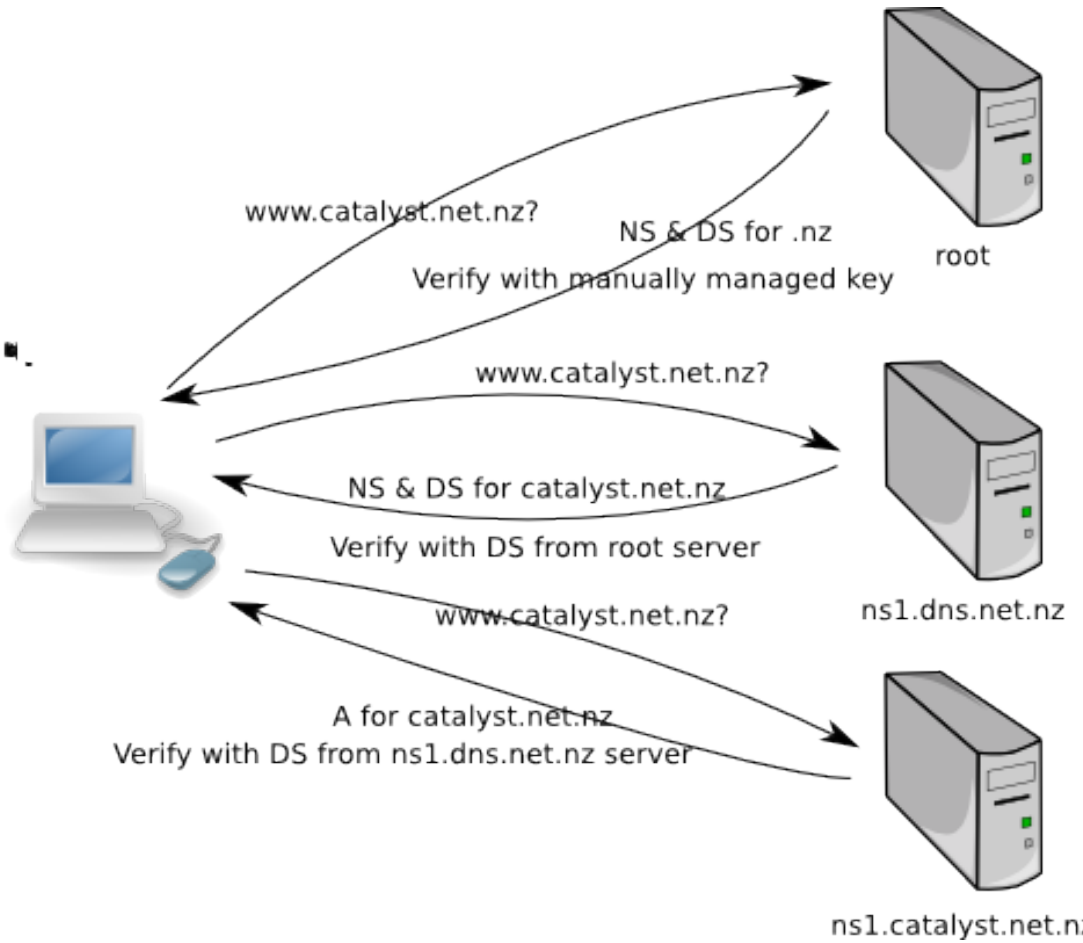
What is wrong?

- UDP 512 byte limit
- Non-existence of TLDs (i.e., `.local`)
- Using UDP, not TCP
- Network providers blocking TCP port 53
- Possible extra load due to IPv6?
- Phishing due to Unicode (IDN)
- No data integrity (beyond IP layer)
- No data origin authentication

Assurance of Trust

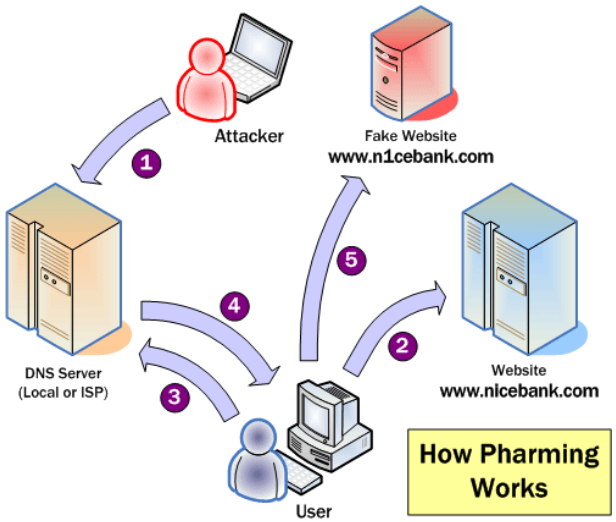
- Build a path of trust to a zone
- That path of trust can have multiple start points - "Secure Entry Points"
- Be assured that the response:
  - hasn't been tampered with;
  - came from an authoritative source
- Ultimately protecting the end-users

Assurance of Trust



Prevent Attack - Name Chaining

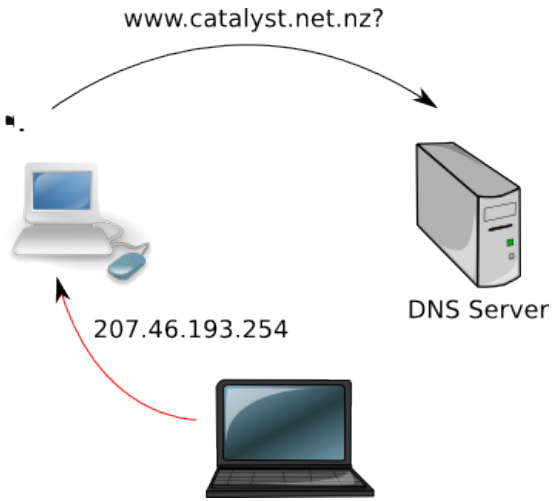
- Commonly known as DNS cache poisoning
- Attack involves arranging for DNS cache on recursive nameservers to have wrong records for a domain name



- In 2006 customers of Rogers and Bell Canada were attacked using this method, The A records for TD Canada Trust and CIBC web servers were poisoned.

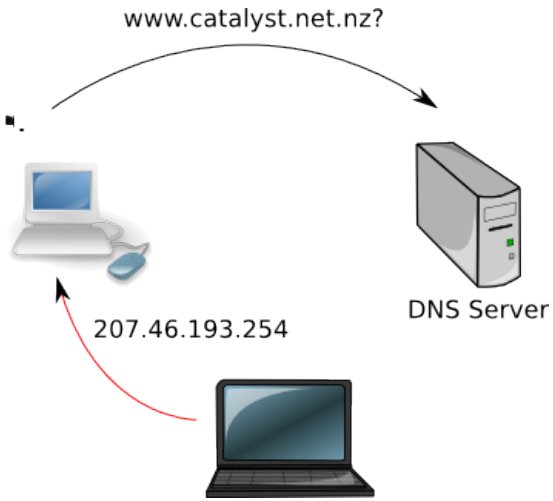
Prevent Attack - Packet Interception

- Man-in-the-middle class attacks
- Active attack, the DNS request packet is intercepted, and a false response is sent
- Requires that the bad guys are able to intercept your packets
- Wireless...



Prevent Attack - ID Guessing and Query Prediction

- Similar to Man-in-the-middle class attacks
- Only a limited number of IDs used in queries
- DNS resolvers use the first response received
- Doesn't have to be on the same network, as with Man-in-the-middle.



Attacks Not Prevented

- Betrayal by Trusted Server
- Denial of Service

Time line

- 1983: Mockapetris invents DNS
- 1986: IETF RFC1034 (Concepts) & 1035 (Implementation)
- 1988: Widespread use
- 1990: Steven Bellovin discovers flaws in DNS, keeps it secret.
- 1995: Flaw is published, IETF starts to talk about DNSSEC
- 1997: RFC2065 - First attempt at DNSSEC
- 1999: RFC2535 - DNSSEC looks finished, but a lot of discussion on parent-child interaction/authority
- 2000: First DNSSEC TLD test bed, .nl.nl shadow zone
- 2001: SECREG.nl experiment - though successful, .nl does not continue
- 2001: NLnetlabs becomes a major developer of NSD, supporting DNSSEC and LDNS library.

Time line

- 2002/2003: RFC2535bis - the DS record introduced
- 2003: Dutch ISP xtdnet.nl enables DNSSEC on all customer domains
- 2005: RFC4033, 4034 & 4035 published - "DNSSEC"
- 2005: Sweden (.se) becomes first TLD to deploy DNSSEC
- 2006: RIPE enables DNSSEC for their in-addr.arpa
- 2007: Deployment worldwide increased to 5 TLDs (Bulgaria, Puerto Rico, Brazil, Midway)
- 2007: Opt-in still being discussed to reduce memory requirements in large zone files
- 2007: Large TLDs still working on faster hardware and protocol tweaks to be able to sign their zones daily (or in some cases hourly)

History

- IETF open design team meeting, November 1993
- Desired security services:
  - Data integrity
  - Data origin authentication
- Excluded:
  - Encrypting requests/responses - DNS is published
  - Authentication of clients/servers for access control
- Decided to use a digital signature mechanism
- While .se is first TLD to enable DNSSEC in 2005, there is a catch...

Benefits

- Increase reality of security
- Trusted DNS provides breeding ground for applications
- Allows error detection
- Trust for moderated domains:
  - `.govt.nz`, `.gov.au`, others

Components

- **Signers** - Actually signs the records in the zone
- **Authoritative Nameservers** - Publish the signed zones, doesn't perform any crypto
- **Recursive Nameservers** - Query the signed records, might verify the DNSSEC signatures
- **Local Resolver** - The DNS resolver on the computer making the query, might verify the DNSSEC signatures
- **System API** - Provide proper feedback to the applications, not just ServFail or "not found"
- **Applications** - If the DNSSEC validation failed (and the API told us), inform the user

Extra Resource Record Types

- **DNSKEY** - DNS Public Key for the private key used to sign the zone.
- **RRSIG** - Resource Record Signature, digital signature for a RRset in the zone.
- **DS** - Delegated Signer, hash of the public key for a DNSSEC secure child zone, similar to NS "glue" records.
- **NSEC** - Next Secure, pointer to the next entry in the zone. Provides authenticated denial of existence.

RR Type - DNSKEY

- DNSKEY

```
live.etc.gen.nz. 3600 IN DNSKEY 256 3 5 (
    AwEAAeWW7W6jCTbE7hbmYEyVP37cjTh8lSrnZ9CIj5p0
    MEG3dReTEtuXIiE/l8K6HjkuVZ+tRyrW8IKO58hxDd/b
    vEsKB8nwvlu625FZCA+j8qazjsO/WNUUKeXqC6oy/bBP
    /z2pAw9OgXYK68hSUSkNZT4mjSd94WhCeAp3XSx8NHGt
) ; key id = 16260
```

RR Type - RRSIG

- RRSIG for an AAAA

```
www.nlnetlabs.nl. 504 IN AAAA 2001:7b8:206:1::1
www.nlnetlabs.nl. 504 IN RRSIG AAAA 5 3 600
  20080220104240 (
    20080123104240 18182 nlnetlabs.nl.
    bpwhw5Pn7DVQEZsgdyOMBjfvSOeYgNf16X5jgm/+jkSy
    9N9cKxY7ApjrDKbnGwpH+G6ufFNZKvZZfpVjOnGfLdlg
    tW8gpv2KnBOfNQj7cuskSMmqDMcwgzdl1A/iGGtuqrFF
    mnjyNH6z/z8O9dLKntg05U6EMOVr5pu8Lv0YPwM=
  )
```

RR Type - DS & NSEC

- DS

```
live.etc.gen.nz. 3600 IN DS 29713 5 1 (
  AAB51DAF4B0D46EA7D2394E4E7B15FBA7CC4D951 )
```

- NSEC

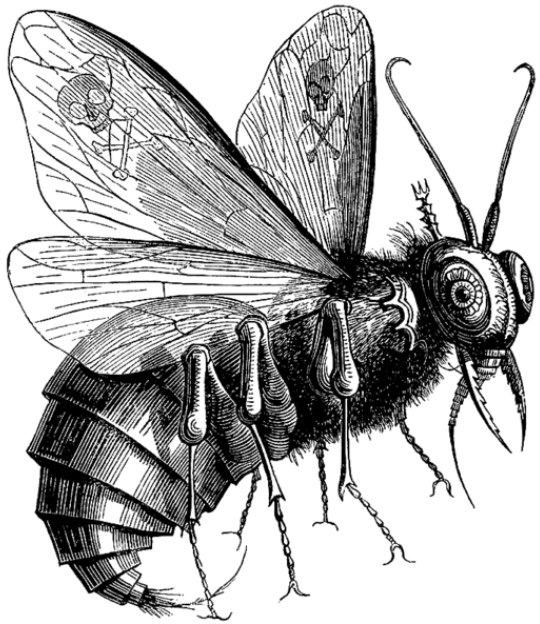
```
live.etc.gen.nz. 3600 IN NSEC
  cerberus.live.etc.gen.nz.
  NS SOA TXT RRSIG NSEC DNSKEY
```

Authenticated Denial of Existence

- Hold on, why have a pointer to the next record?
- We want to notify the users that a record doesn't exist, with a signed response.
- Could sign responses on the fly - burn that CPU!
- NSEC allows these records to be pre-generated during the signing stage.

Authenticated Denial of Existence - Example

- What if I query `beelzebub.live.etc.gen.nz`?



Authenticated Denial of Existence - Example

- What if I query `beelzebub.live.etc.gen.nz`?
- Get the **signed** NSEC record for `live.etc.gen.nz` pointing to the next valid record.

```
live.etc.gen.nz.     5 IN SOA live.etc.gen.nz. root.etc...
live.etc.gen.nz.     5 IN RRSIG SOA 5 4 5 ...
live.etc.gen.nz.     5 IN RRSIG SOA 5 4 5 ...
live.etc.gen.nz. 3600 IN NSEC
  cerberus.live.etc.gen.nz. NS SOA TXT RRSIG NSEC DNSKEY
live.etc.gen.nz. 3600 IN RRSIG NSEC 5 4 3600 ...
```

How does NSEC work?

- Sort the zone file
- Insert the NSEC records after every record, pointing to the next (non-NSEC) record
- Sign every record in the zone file (including NSEC)

Extra DNS Headers

- Requires:
  - **EDNS0** - Extended DNS, allows adding more flags and options in DNS packets
- Adds:
  - **CD** - Checking Disabled, specify that you don't want DNSSEC validation to occur
  - **AD** - Authenticated Data, that the DNS response has DNSSEC validated data

Signing the Root - Trust Anchor

- Politics
- Who to trust?
- Can break an entire TLD in a blink of an eye
- Stop-gap measures:
  - DNSSEC Lookaside Validation (DLV)
  - Manually manage keys
- Solutions?
  - Some proposals out there
  - Disagreement about best way forward

Signing the Root - DLV

- People want to start using DNSSEC yesterday
- Even in TLDs that aren't DNSSEC enabled, and won't be for ... a while.
- Let's use a database listing the DNSSEC domains, and their DNSKEY.
- Resolvers check for DNSSEC in the TLD first, if that fails, query our database.
- The database has to be as reliable as the DNS.
- The database has to be as secure as DNSSEC.

Signing the Root - DLV

- Sounds like a job for the DNS!
- Store a hash of the DNSKEY in another zone with DNSSEC enabled.
- If `gen.nz` doesn't have DS for `etc.gen.nz`
- Then look for DLV in `etc.gen.nz.dlv.isc.org`
- Yes, another new Record Resource type.
- Manually add (and manage) the DNSKEY for dlv.isc.org as a Trust Entry Point.
- Issues:
  - Have to trust a 3rd party - ISC.
  - ISC have to validate entities submitting DLV records.

Zone Walking - NSEC

- More politics!
- NSEC allows zone walking
- A number of TLDs have policies which disallow zone transfers, which zone walking allows
  - .uk, .de, .nz, others
- Some TLDs don't care

Zone Walking - NSEC

```
switchblade:~$ ldns-walk nlnetlabs.nl @ns1.nlnetlabs.nl
nlnetlabs.nl. A NS SOA MX TXT AAAA NAPTR RRSIG NSEC DNSKEY
_sip._udp.nlnetlabs.nl. SRV
alpha.nlnetlabs.nl. A MX AAAA RRSIG NSEC
bartok.nlnetlabs.nl. A MX AAAA RRSIG NSEC
behemoth.nlnetlabs.nl. A AAAA RRSIG NSEC
conch.nlnetlabs.nl. A AAAA RRSIG NSEC
dhcp-01.nlnetlabs.nl. A RRSIG NSEC
dhcp-02.nlnetlabs.nl. A RRSIG NSEC
dhcp-03.nlnetlabs.nl. A RRSIG NSEC
...
switchblade:~$
```

Zone Walking - NSEC3

- NSEC3 IETF proposed standard as of December 2007.
- Still sorts the zone first, but now uses a hash of the domain name.
- New Resource Record types:
  - NSEC3 - Hash of the next secure record
  - NSEC3PARAM - NSEC3 parameters that were used to calculate hashes
- Software support (not production code):
  - BIND, NSD, LDNS, others...

Key Roll Over

- Crypto keys should be replaced regularly
  - Reduces chance they'll be broken
  - Provides procedure if they are compromised or lost
- DNS has a time to live (TTL), used to determine how long to cache data
- Roll over has to allow for this, as old key could hang around for a while
- Solution - Publish two keys

Key Roll Over - Procedure

- Zone publishes current DNSKEY (A), plus next DNSKEY (B)
- Parent publishes current DS (A)
- *...time passes...*
- Zone publishes old DNSKEY (A), plus current DNSKEY (B)
- Parent publishes current DS (B)
- *...time passes...*
- Zone publishes current DNSKEY (B), plus next DNSKEY (C)
- Parent publishes current DS (B)
- The amount of time that passes depends on TTL of RRSIG and DNSKEY records
- Requires that RRSIGs are signed with both old and current DNSKEYs.

Key Roll over - Simplify?

- Reduce requirement to have all RRSIGs signed with both old current DNSKEYs
- Reduce requirement to change DS records often
- Have two DNSKEYs:
  - Key Signing Key (KSK) with a published DS;
  - Zone Signing Key (ZSK), which is signed by the KSK.
- The KSK is used to actually sign the zone
- Typically the KSK is a stronger key, and has a longer life than the ZSK
- Reduces requirement to change published DS
- Spends up zone signing as ZSK has a shorter length

Application Support

- What use is DNSSEC if the end-user isn't alerted to issues?
- Applications have to support DNSSEC
- Patches available for at least Postfix, Sendmail, OpenSwan
- Simple extension for Firefox, shows green tick for authenticated hosts
- How should we handle failed lookups?
  - Failed check;
  - No chain of trust
- Most users just click through SSL certificate warnings...
- Application support increases the feeling of security

Router Support

- Not just applications, hardware support too!
- Consumer broadband router test by Patrik Wallstrom from Sweden in December 2007
- 10 routers tested, only *1* passed!
- Partly due to:
  - EDNS0 header
  - Larger packets
  - New RR types
  - Queries over TCP
- Major issue! A show-stopper?

Router Support

- Not just applications, hardware support too!
- Consumer broadband router test by Patrik Wallstrom from Sweden in December 2007
- 10 routers tested, only *1* passed!
- Partly due to:
  - EDNS0 header
  - Larger packets
  - New RR types
  - Queries over TCP
- Major issue! A show-stopper?

Adoption Map

This map was created by Paul Wouters

Reasons for Slow Adoption?

- Politics!
- Root zone signing?
- Zone walking (in some cases)
- TLD registries don't think it is required?
- Registrants aren't demanding it.
- Zone file size increase

Future Applications

- DNSSEC not designed to provide secure storage of data, but...
  - SSH host keys (RFC4255)
  - SSL/TLS public keys
  - Opportunistic VPNs (for IPsec: RFC4025 & RFC4322)
  - GPG public key deployment
  - ENUM
  - Domain Keys Identified Mail

Summary

- DNSSEC - It works ... mostly.
- Technically feasible, but will it actually happen?
- Hardware support could take years to sort out.
- Transition period could reduce the reality and feeling of security.
  - Depends on how applications handle failure conditions.

More Info

- http://www.dnssec.org
- How To: http://www.nlnetlabs.nl/dnssec_howto
- DNS History: http://nlnetlabs.nl/dnssec/history.html
- Steven Bellovin: http://www.cs.columbia.edu/~smb/papers/dnshack.ps
- RFC3833 - Threat Analysis of the Domain Name System
- NSEC3: http://www.ietf.org/internet-drafts/draft-ietf-dnsext-nsec3-13.txt
- Map: http://www.xelerance.com/dnssec/
- DLV: http://dlv.isc.org
- DNS Cache Poisoning: http://palisade.plynt.com/issues/2006Mar/pharming/

Even More Info

- DNSSEC with NIST: `http://www.x-antd.nist.gov/dnssec`
- DNS Operations (DNSOPS): `http://www.ietf.org/html.charters/dnsop-charter.html`
- Apster (APNIC newsletter for Asia Pacific region): `http://www.apnic.net/docs/apster/index.html`
- dnssec-deployment mailing list: `http://mail.shinkuro.com:8100/Lists/dnssec-deployment/`

Questions

- Any questions?

Some Answers

- Why not just use emptyish signed packets that say "record doesn't exist" instead of NSEC?
  - Authoritative nameservers don't perform any crypto, so would have be a generic, pre-generated response.
  - Which means a bad guy could capture it, then send it back to anyone to pretend a domain name doesn't exist, bad.
- In NSEC, why not point to something like hostname . "GARBAGE" (so, `cerberusGARBAGE`)?
  - This doesn't provide authenticated denial of existance for domains between `cerberus` and `chimaera`.
  - As a bad guy, I'd just send back the NSEC record for `cerberusGARBAGE` if I wanted to hide `chimaera`, the resolver doesn't know what should come next.

More info on NSEC

- NSEC has been designed so zones can be signed offline.
- If you query a non-existant host, you get the NSEC record for the existing host that would be before it.
  - Query for `beezlebub.etc.gen.nz`, get `live.etc.gen.nz` (start of zone)
- Which points to the next valid host.
  - So `live.etc.gen.nz` points to `cerberus.live.etc.gen.nz`
- Therefore you can receive a validated response proving that a host doesn't exist.
- NSEC means that the servers don't need to do any crypto and responses can be cached.
- We can't just use negative responses anymore, as they aren't, and can't be signed (since we don't do crypto on the servers).

Some More Answers

- You said that DNSSEC will make the response bigger, can you quantify that?
  - Quick comparision, using dig to lookup www.etc.gen.nz.
    - No DNSSEC: 184 byte response
    - DNSSEC (using +dnssec option): 890 byte response
  - So expect to see a significant increase in response size
  - A quick search using Google for "dnssec response size increases" returned a large number of results, the first page at least looks interesting.
- If I enable DNSSEC for my zone, will all queries be bigger?
  - As you can see in the previous example, no. The DNSSEC records are only sent if they have been asked for, normally by a resolver that is performing DNSSEC validation.

Thank You

- Andrew Ruthven
- Catalyst IT Ltd
- andrew.ruthven@catalyst.net.nz