An Overview of DNSSEC

Olaf M. Kolkman
olaf@nlnetlabs.nl

NLnet Labs

About me

- Director of NLnet Labs, a charity working on open standards and open source software
  - NSD, Unbound, ldns, Net::DNS, Net::DNS::SEC, DNSSEC evangineering (training, documentation)
  - Routing & Addressing research
  - IETF, ICANN, and various advisory roles
- Previously @ RIPE NCC: responsible for DNSSEC deployment
  - DNSEXT chair 2001-2006
  - IAB member since, 2006. Chair since 2007

NLnet Labs

DNS

- Domain Name System
- Provides the mapping from names to resources
- A global, distributed, loosely coherent system
- Almost all transactions on the Internet use the DNS

NLnet Labs

DNS has a distributed nature

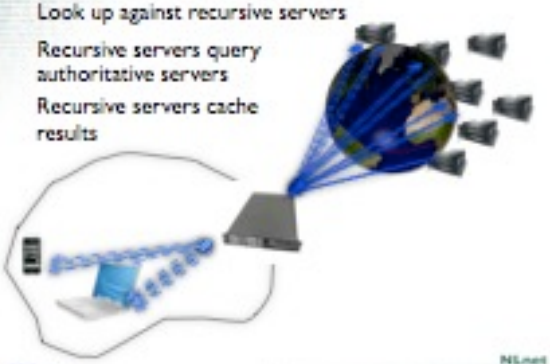
- Authoritative servers all provide part of the name space
- User devices query a local server that maintains a cache
  - For better performance
  - For scalability of the system as a whole

Animation

Look up against recursive servers

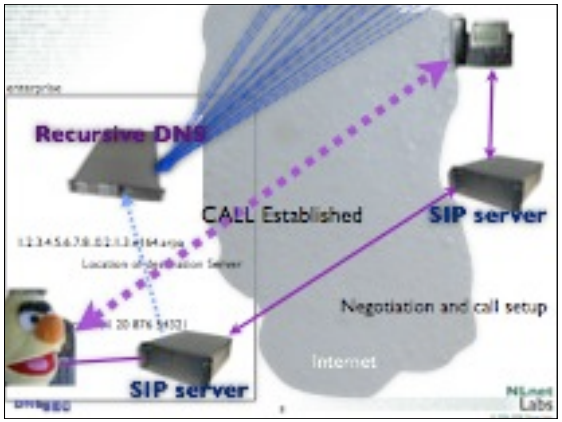
Recursive servers query authoritative servers

Recursive servers cache results

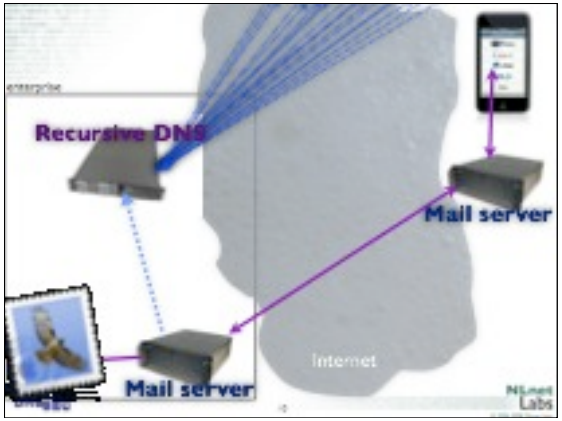


When do you use the DNS

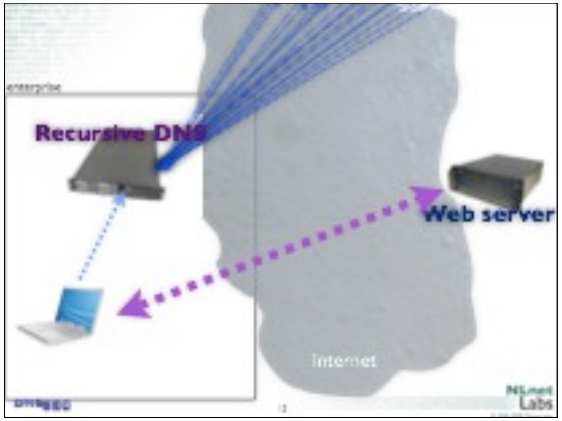
- Anytime that you need to know where the other guy is
- DNS is the phone book of the Internet
- So it is used when people make a voice over IP call



Or they use the DNS when sending MAIL



Or they use the DNS when browsing the Web



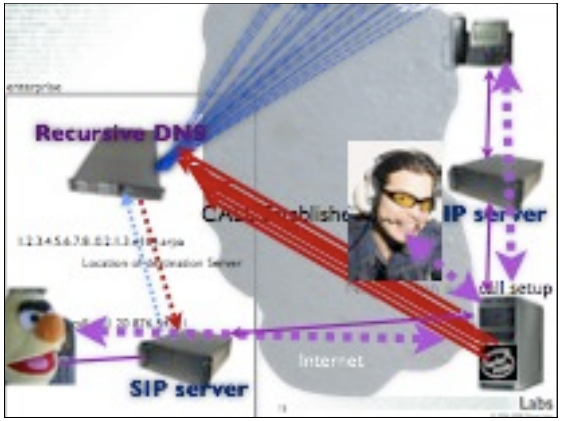
Or they use the DNS

- When downloading Software upgrades
- Sharing their agenda
- Uploading tax forms
- Instant messaging with friends
- Connect to their security camera
- Figure out the latest news about that merger

So DNS is IMPORTANT

- How would an attacker use the DNS for attacks?
- By fooling the receiver that a service lies elsewhere

Back to our VOIP example



Cache Poisoning

- The attack you just saw is called cache poisoning
- Inserting false data into the cache of recursive name servers
- This form of attack has been known for years
- One of the reasons to work on DNSSEC

Cache poisoning is a generic attack

- If somebody manages to replace DNS data they can hijack the service that DNS record pointed to.
  - Voice, web services, online banking and tax, anything!

Why is this attack possible?

- Attack is based on "predicting" properties
  - e.g. when asking a question to a female you expect a female voice to answer
  - The property traditionally available is the Query ID

First: randomness is non-trivial

- BIND used a pseudo random number generator that provided predictable sequences
  - Current ID even: next ID one out of 10 possible numbers
  - Only order 15 queries needed to predict rest of the stream
- Discovered by Amit Klein of trusteer

Second: Query ID only is not sufficient

Chance that *n* people have different birthdays

$$p(n) = 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{n-1}{365}\right) = \frac{365 \times 364 \cdots (365 - n + 1)}{365^n} = \frac{365!}{365^n (365 - n)!}$$

Chance that *n* people have the same birthday

$\bar{p}(n) = 1 - p(n)$.

n	P(n)
0	1 (100%)
10	0.883 (11.7%)
20	0.589 (41.1%)
30	0.294 (70.6%)
40	0.109 (89.1%)
50	0.030 (97%)
60	0.006 (99.4%)
70	0.001 (99.9%)

Varying properties in a packet as defence

- The sender can vary the following properties for the attacker to match
- Query ID (16 bits)
- Source port (16 bits)

Using all ports, not easy

- Some architectures did not use a sufficiently large range of ports
- The patches issued early 2008 all had to do with increasing the randomness in port use

Bits	50%	5%	Aka
16	10 s	1 s	Unpatched server, random ID
26	2.8 h	17 m	Patched, using only 1024 ports
34	28 days	2.8 days	unbound with defaults
44	28444 days	2844.4 days	unbound with 0x20 and source addresses configured

0x20?

- Variables in a query:
- qname, qtype, qclass, **query-id**, ip-source-address, **ip-source-port**
- 0x20 uses query name:
  - BlaFoO.SeCurE-BanK-ServiCeS.COm
  - 10010110100101001010000101110

50%-5%-0.5%-0.05%

- There are literally millions of resolvers out there
- The calculations are based on certain assumptions
  - Scanning of Port ID and Query ID are independent: multiplication of chances?
- All steps in an arms run, do we count on the next quick fix or the solution that has been designed to cope with this?

Still: until 2007 folk seemed happy

- Attacker only got one try:
  - Query for www. onlinebank.example
  - Bombard with answers hoping for the the mala-fide answer to get in first
  - Wait for timeout of the TTL
  - Then try again

Kaminsky's variant

- Classic cache poisoning gave you 'a few tries' to get in between the outgoing question and incoming answer
- Kaminsky came with a scheme where the culprit can keep trying
  - Surprisingly simple, a wonder nobody thought of the variety before

Oopss

- Query:
  <randomcruft>.www.importantbank.example
- respond with fake delegation to:
  www.importantbank.example with glue
- There are other varieties too, but this is the one that has no real workaround

problem?

There is Recognition

http://www.kb.cert.org/vuls/id/800113

There is Exploit Code

And more exploit code

http://service.net/cv/~s0rd/archive/dns/

The networks are scanned

http://asert.arbornetworks.com/2008/07/10-day-of-dns-crack-activity/

There have been succesful attacks

http://isc.sans.org/diary.html?storyid=1801

Consider turning on your sniffers

- In all honesty: not much is known about how much and how often success is accomplsihed
- See wether there is attack traffic around
- Look for spoofing attempts of software update domain
- Share your experience

DEMO !?!

We lost DNS...
How about the other defenses ?

SSL?

- Current practices are sloppy
- Users connect to their banks
- Get redirected to unrelated domains
- User interfaces only show padlocks

For example

Exploit

- Attacker poisons DNS for www.postbank.nl
- Fake www.postbank.nl redirects to postbank.secure-bank-services.com
  - Obtaining the domain name and certificate is trivial for organized criminals
- Users are used to these sort of redirections and the domainname looks trustworthy

Things get worse

- Fake www.postbank.nl redirects to fake https://www.postbank.nl
- SSL protects agains that?
- Not if the attacker has a signed certificate
  - How would an attacker do that?

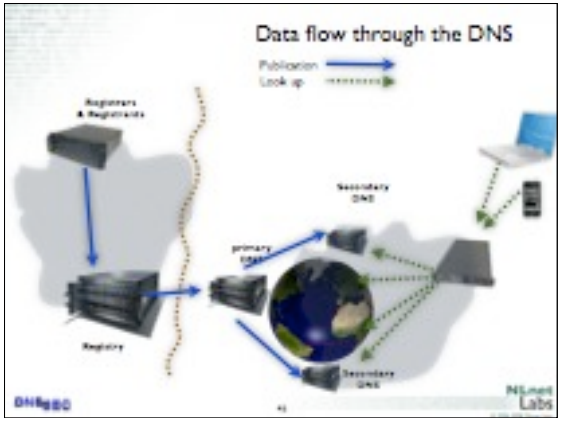
Don't rely on DNS for the Security review

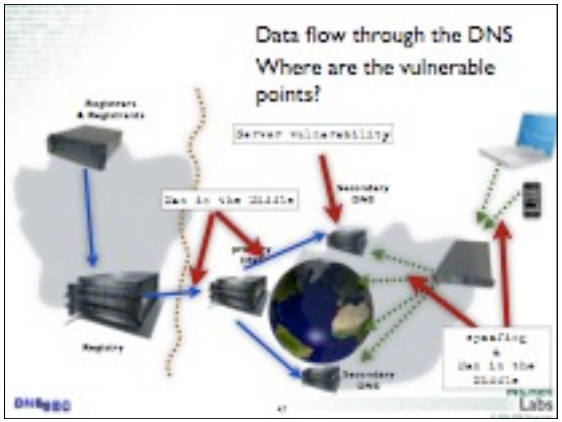
- Don't get the contact details out of the WHOIS, getting to WHOIS is DNS based
- Don't send confirmation e-mails to typical addresses in the domain
  - Mail uses the DNS
- Don't try to see if domain already has a SSL certificate installed. That uses the DNS

Introducing

DNSSEC
What does it protect?

Let us have another look at the DNS architecture





DNSSEC protects all these end-to-end

- As an aside:
  There is a protection mechanism against the man in the middle: TSIG
  - Provides hob-by-hop security
  - TSIG is operationally deployed today
  - Based on shared secret: not scalable

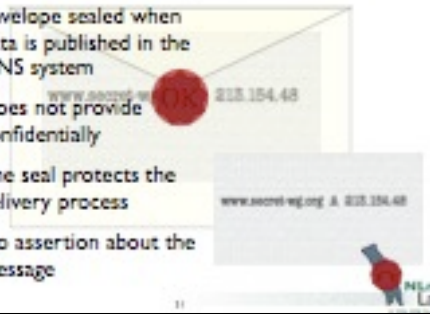
What does DNSSEC provide

- provides message authentication and integrity verification through cryptographic signatures
  - You know who provided the signature
  - No modifications between signing and validation
- It does not provide authorization
- It does not provide confidentiality
- It does not provide protection against DDOS

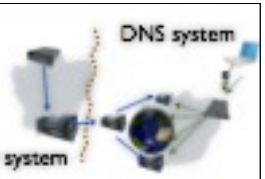
Metaphor

Metaphor

- Envelope sealed when data is published in the DNS system
- Does not provide confidentially
- The seal protects the delivery process
- No assertion about the message



Trust and Confidence



- DNSSEC enables confidence in the DNS
- It does not change the trust we put in the Registry/Registrar procedures
  - Although introduction of DNSSEC may improve some of the procedures

State of DNSSEC Deployment

The Numbers

(A sad state of affairs)

- http://secspider.cs.ucla.edu/ reports a little over 10,000 zones signed, only little under 1000 are production zones
- Reverse zones in the RIPE region
- .se, .pr, .br and .bg are signed top level domains
- .gov, .uk, .arpa, .org, .in, and a few enum trees have voiced some form of commitment
- There is a testbed for the root and there has been a 'notice of interest' from NTIA

Chicken and Egg

- Little deployment means little experience and few tools.
- Little experience and few tools increase the cost of deployment
- Little signing infrastructure to justify cost of validation
- Little validators to justify the signing infrastructure
- No short term benefits, only long term
  - This is about Trust in the Internet

Software and Tools

- Open Source
  - BIND, NSD, and Unbound
  - Various maintenance tools
    - http://www.dnssec.net/software
- Vaporware
  - opendnssec.net and many more
- Commercial
  - Secure64, Xelerance, Infoweapons

http://blogs.technet.com/samdrake/archive/2008/10/30/dnssec-in-windows-7.aspx

Breaking the Egg

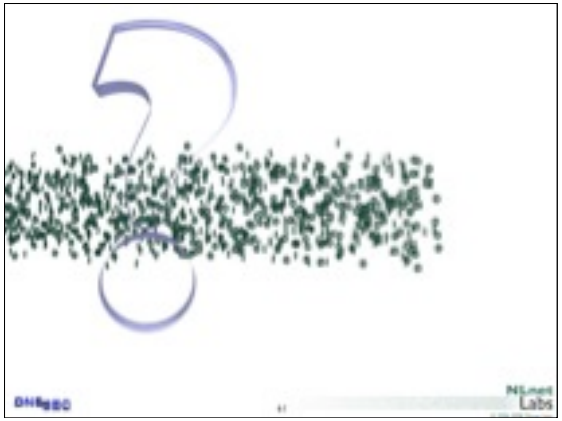
- Deployment by the custodians of the DNS infrastructure (TLDs and the Root) allows others to hook in
- Resolver side deployment to immediately benefit

Suppose

- a TLD registry, some registrars, major ISPs, and some major stakeholders like banks, online services, and public services would simultaneously commit to a schedule?
- That would create immediate value (higher security) for all parties.

Conclusion

- DNS is core to most interactions on the internet
- It is designed with a trust model in mind that has not held for over a few decades
- DNSSEC is a key to (re?)building trust in the Internet
  - Not a magic bullet, but a (necessary) tool



Steps towards Secure DNS

Various Players

- Suppose we want confident mapping for www.bank.in
- What needst to be done

From the resolvers view

- The resolver will need to verify the signature over www.bank.in is valid
- Two tasks:
  - implement a verifying recursive nameserver
  - configure the appropriate public key
  - maintain the configured public keys

DNSSEC on a Recursive Nameserver

- Install the appropriate piece of software
- Latest BIND or Unbound
  - Both run on commodity hardware
  - Both are open-source freely available
- Perform the appropriate amount of testing to understand the failure modes

Configuring Public Keys

- Public keys are configured in the files (manual)
- Make sure public keys are rolled
- Make sure you know the policies of the signing entities
- Not rolling turns into severe failure mode
- Use tools
  (RFC 5011 implementation will be available from NLnet Labs shortly)

The costs involved

- Commodity hardware
  - No need to upgrade Routers and IP equipment
  - Though Firewalls may cause issues
- Free software
- The rest is a knowledge exercise

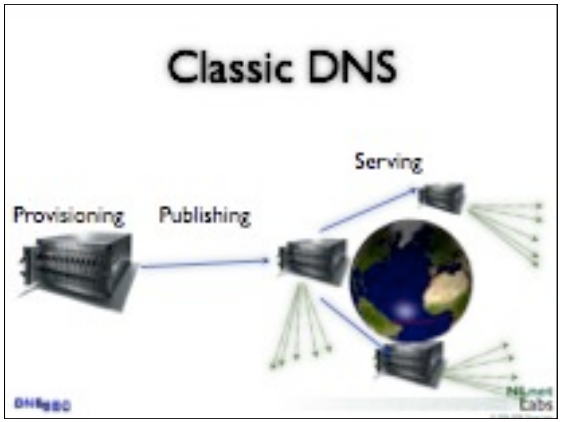
Public Keys

- Suppose you want to verify the data from all your banks:

Use the DNS for public key distribution

- Publish the public key of bank.in in .in
- Have .IN signed
- Reduces the key-maintenance issues greatly!
- Signing .IN facilitates DNSSEC for all parties involved


From the Authoritative end
• A few more components are involved
• Let us zoom in on the several components
• The generic case
• For a zone with delegations (like .IN)


Classic DNS
Serving
Provisioning
Publishing


Introducing DNSSEC
DNSSEC aware name server Software
Serving
Provisioning
Publishing
Key maintenance
&
Zone signing

DNSSEC Aware Nameservers

- Software Exercise
- BIND and Unbound free and open source
- Some hardware requirements:
  - memory requirements increase
  - RIPE 352 or measure

Key maintenance Private Keys

- Determine a policy and implement it
- Think about risks and operations

Risk	On or offline	System consideration
high	on	HSM (FIPS Level 4)
high	off	Reviewed procedures, Physical Safe
medium	on	HSM (FIPS level 2) or shielded system
medium	off	Reviewed Procedures
low	on	Connected or Local system
low	off	System

Key maintenance Public Keys

- Your users may configure your public key as a trust anchor
  - Consider how your users will fetch the key: Out of band validation
  - Document your procedures
- Upload to your parent

Key Maintenance Rollovers

- Document rollover procedures
  - Take into account the timing sequences
  - Understand, train, and automate

Signing

- Use the BIND/Unbound tools
- Depending on your requirements build or buy machinery that allows secure key storage
- Open source tools and proprietary solutions

Note about costs

- Again: Knowledge Exercise
  - Understanding the issues about publication, maintainenance and rolling of the keys
- Draw up requirements
  - Implement or buy solutions
- On the server side: Simple upgrade of software

For the Registry

For the Registry

Questions to address Registrar

- Customer interaction
  - How much checking of your customer setup (value add)
  - How do you validate the public key
    - Is this any different than how you validate a change in the NS?

Questions to addres Registry

- What will you store DNSKEY or DS
  - Consider DS hash algorithm agility: Will you ask all your customers to provide new keys?
- How will you get the DNSKEYs from your Registrars?
  - How is that different from how you get the NS records?

Questions to address Registry II

- What are your operational constraints?
- Will you allow direct Registrant interaction
  - e.g. when a registrants key went broken at 2 am

Follow the NS

- From a registration perspective the NS and the DS data have very similar properties