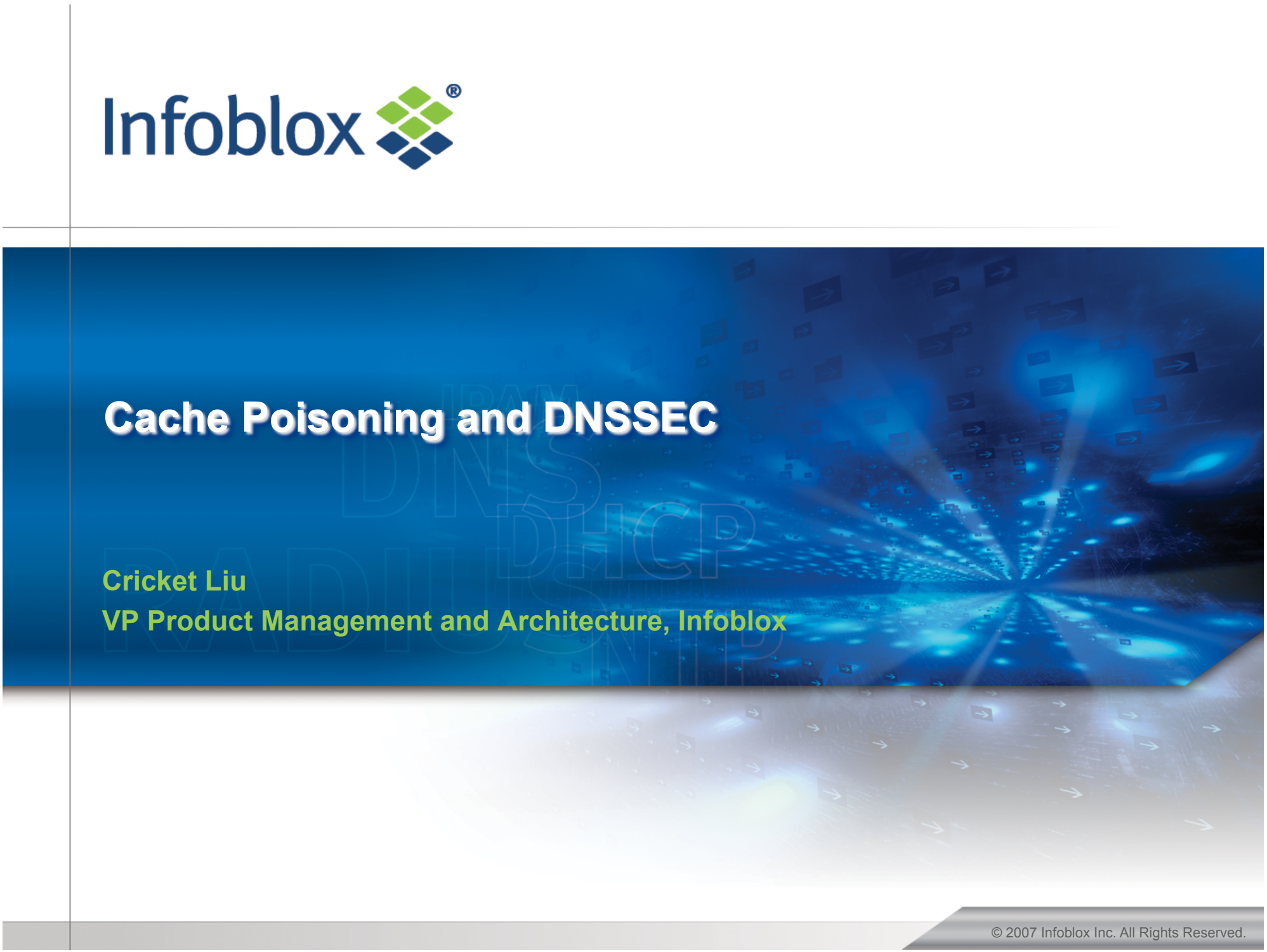
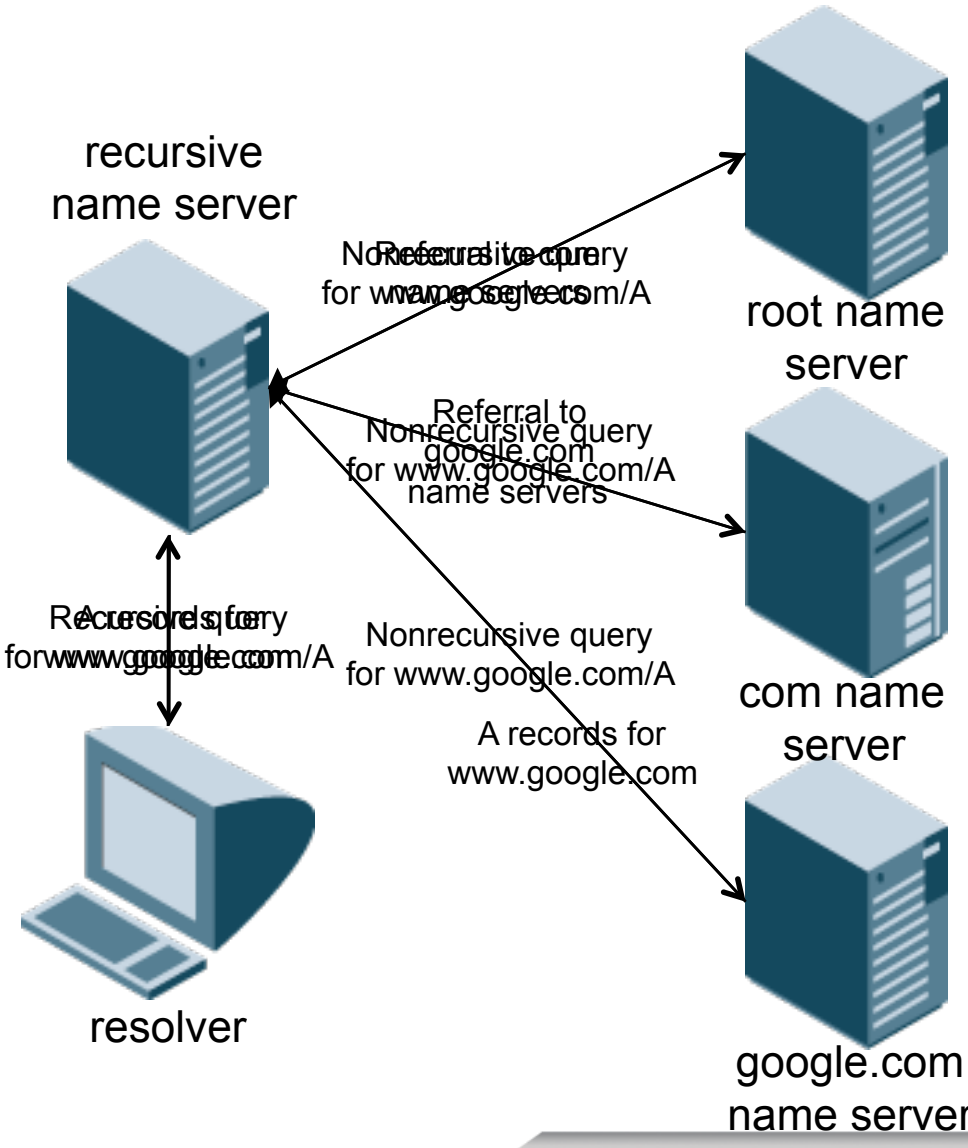


Outline

- A Brief History of Cache Poisoning
- Interim Measures
- The DNS Security Extensions

First, A Review of Recursion

- **What's recursion, anyway?**
  - DNS queries are either recursive or nonrecursive
  - A recursive query asks the name server to do whatever work is necessary to find the answer, including sending additional queries



A Review of Recursion (continued)

Infoblox

- **Who needs recursion?**
  - Stub resolvers (like the one on your computer) send recursive queries to name servers
  - Name servers usually send nonrecursive queries to other name servers
- **Who offers recursion?**
  - External authoritative name servers shouldn't offer recursion at all
    - Assuming they're dedicated to that function
  - Forwarders should only offer recursion to your own internal clients
    - Or even only your own internal name servers

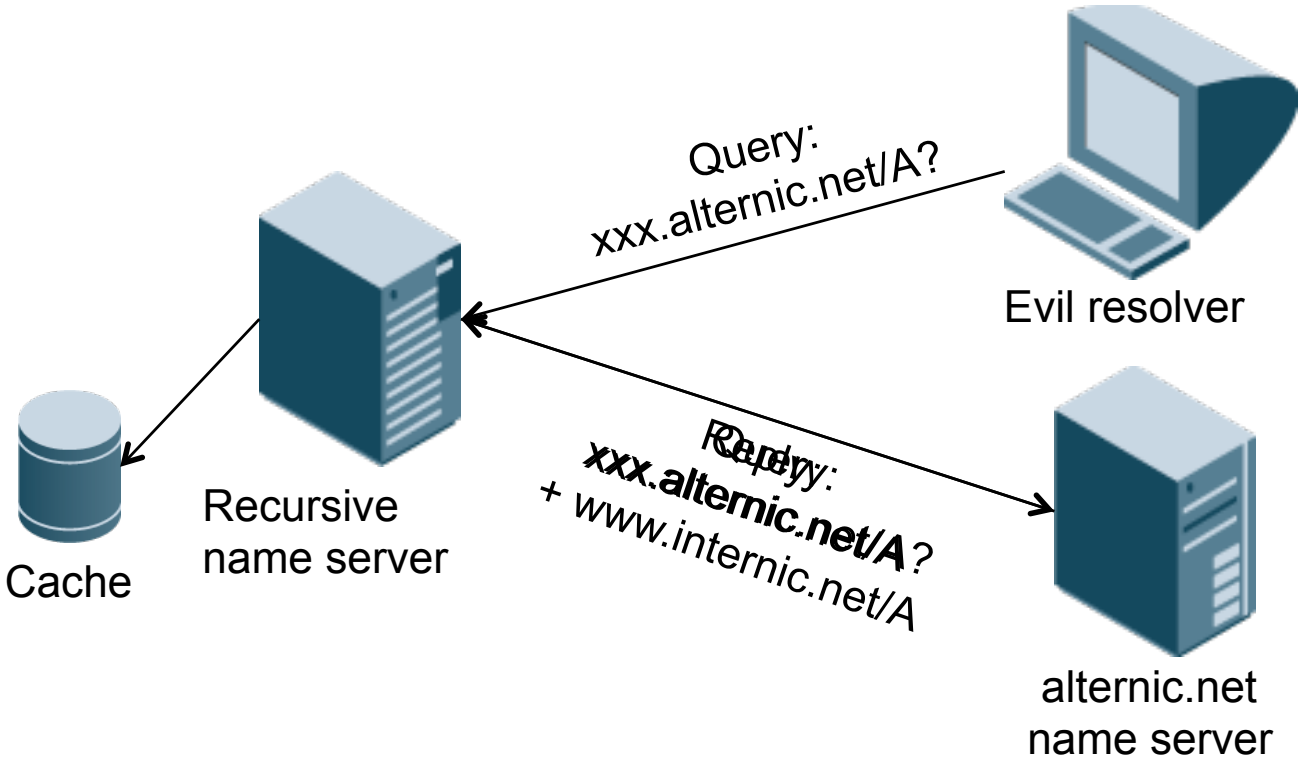
A Brief History of Cache Poisoning

Infoblox

- **What is it?**
  - Inducing a name server to cache bogus records
- **Made possible by**
  - Flaws in name server implementations
  - Short DNS message IDs (only 16 bits, or 0-65535)
- **Made easier on**
  - Open recursive name servers
- **What's it matter?**
  - A hacker can induce your name server into believing something false
    - By caching bogus records
  - Your users might connect to the wrong web site and reveal sensitive data (passwords, account numbers) there
    - The "wrong" web site might look just like the real web site
  - Your users email might go to the wrong destination
    - Where it might just sit, or it might be copied or modified and then sent on

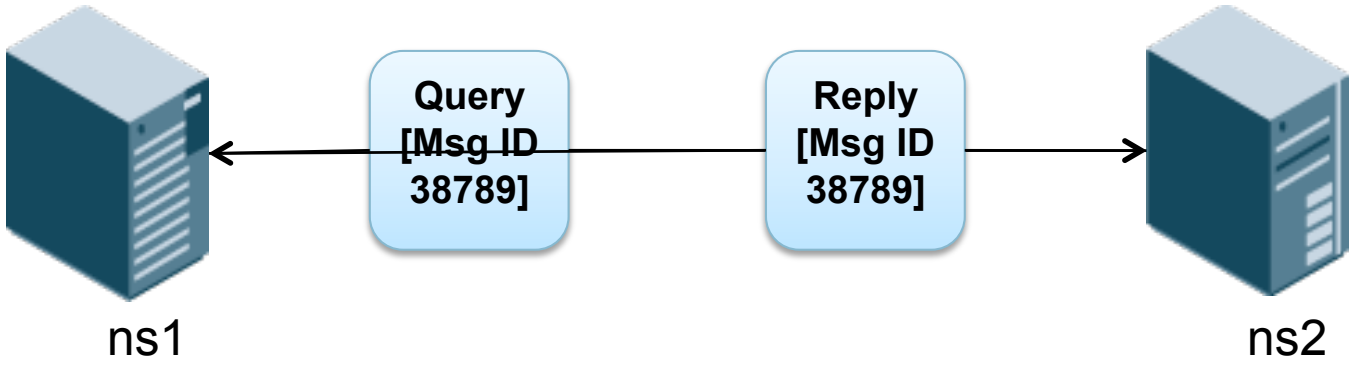
The Kashpureff Attack

- **Eugene Kashpureff's cache poisoning attack used a flaw in BIND's additional data processing**
- **Here's how the Kashpureff attack worked:**



DNS Message IDs

- **A DNS axiom:**
  - The message ID in a reply must match the message ID in the query
- **The message ID is a "random," 16-bit quantity**



True Randomness

How Random--Not!

Infoblox

- **Amit Klein of Trusteer found that flaws in BIND's message ID generator (PRNG) mean that most versions of BIND don't use sufficiently random message IDs**
  - If the current message ID is even, the next one is one of only 10 possible values
  - Also possible, with 13-15 queries, to reproduce the state of the PRNG entirely, and guess all successive message IDs

Birthday Attacks

Infoblox

- **Barring a man in the middle or a vulnerability, a hacker must guess the message ID in use**
  - Isn't that *hard?*
  - As it turns out, not that hard
- **Brute-force guessing is a birthday attack:**
  - 365 (or 366) possible birthdays, 65536 possible message IDs
  - Chances of two people chosen at random having different birthdays:

$$\frac{364}{365} \approx 99.7\%$$

  - Chances of n people (n > 1) chosen at random all having different birthdays:

$$\bar{p}(n) = \frac{364}{365} \times \frac{363}{365} \times \ldots \times \frac{366 - n}{365}$$

  - Chances of a "birthday collision": $p(n) = \left(1 - \bar{p}(n)\right)$

Birthday Attacks (continued)

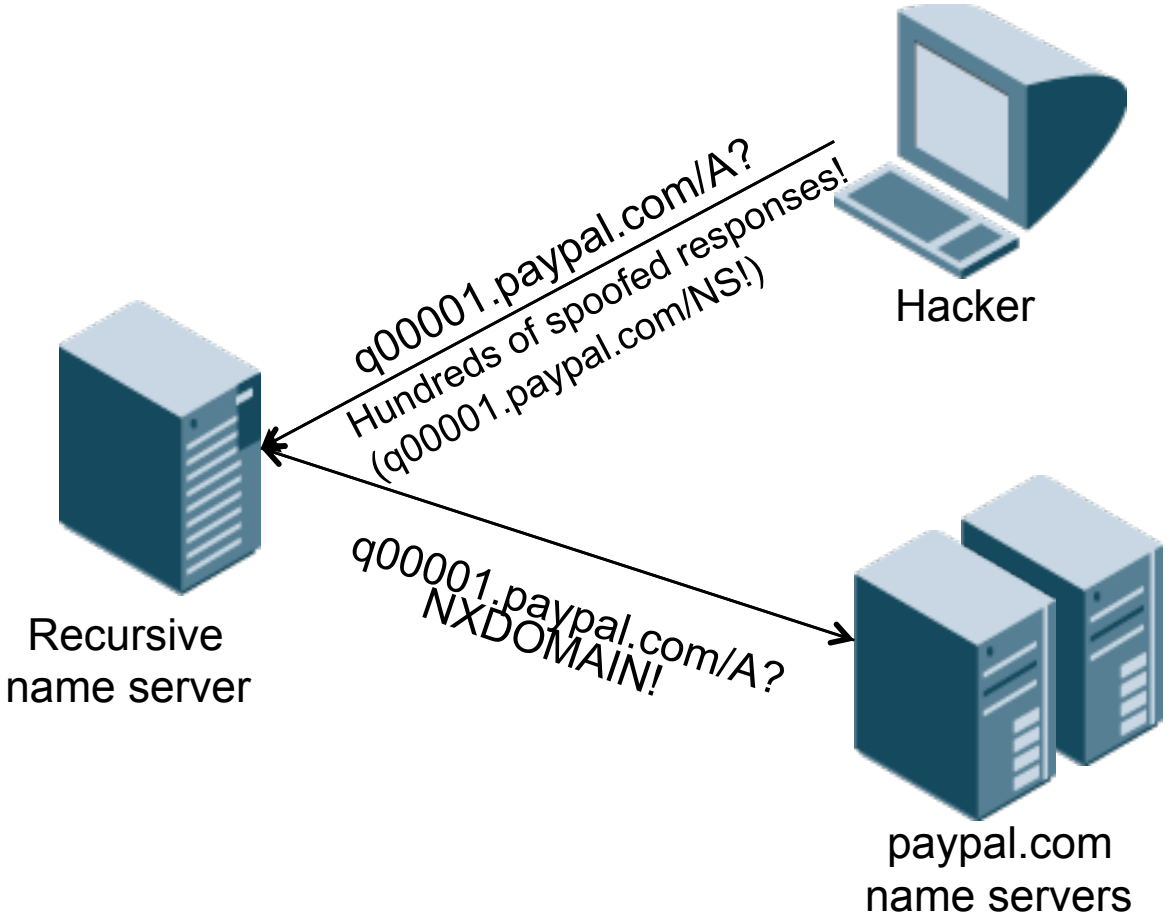
Infoblox

People	Chances of two or more people having the same birthday
10	12%
20	41%
23	50.7%
30	70%
50	97%
100	99.99996%

Number of reply messages	Chances of guessing the right message ID
200	~20%
300	~40%
500	~80%
600	~90%

The Kaminsky Vulnerability

- **How do you get that many guesses at the right message ID?**



The Kaminsky Vulnerability (continued)

- **How does a response about q00001.paypal.com poison www.paypal.com's A record?**
- **Response:**

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61718
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1,
ADDITIONAL: 1

;;; QUESTION SECTION:
;q00001.paypal.com.          IN      A
;;; AUTHORITY SECTION
q00001.paypal.com.          86400   IN      NS      www.paypal.com.
;;; ADDITIONAL SECTION
www.paypal.com.             86400   IN      A       10.0.0.1
```

Saved by the Second Law of Thermodynamics

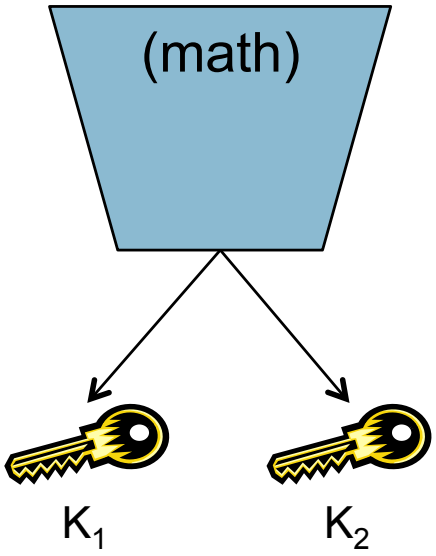
Infoblox

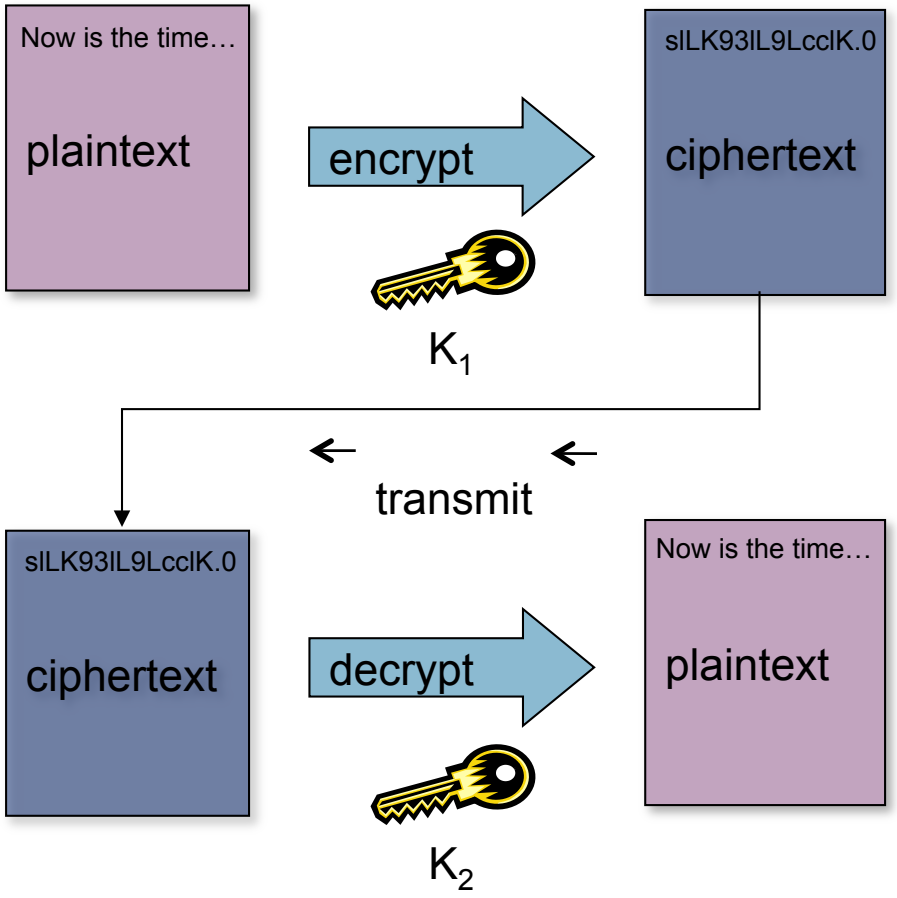
- **To make it more difficult for a hacker to spoof a response, we use a random query port**
  - In addition to a random message ID
  - If we use 8K or 16K source ports, we increase entropy by 13 or 14 bits
  - This increases the average time it would take to spoof a response substantially
- **However, this is not a complete solution**
  - Spoofing is harder, but still possible
  - Evgeniy Polyakov demonstrated that he could successfully spoof a patched BIND name server over high-speed LAN in about 10 hours

The DNS Security Extensions

- **The DNS Security Extensions, or DNSSEC, use asymmetric cryptography to "digitally sign" DNS zone data**
- **This provides**
  - Authentication of DNS data ("Was this data signed by the administrator of the zone?")
  - Integrity checking of DNS data ("Is this the same data that was signed by the administrator of the zone?")
- **To accomplish this, DNSSEC**
  - Adds several new record types
  - Defines new bits in the header of the DNS message (not covered here)
  - Defines the process of validation, which recursive name servers can use to verify signed data
  - Requires new administrative processes

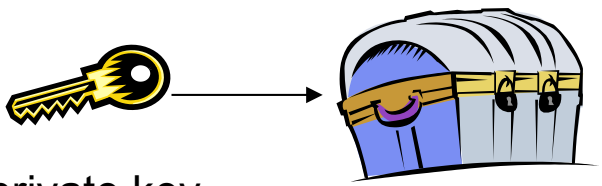
Public Key Cryptography Illustrated



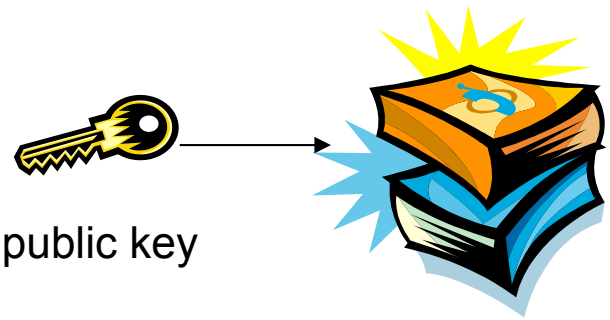


The Key Pair: Public and Private

- **In practice**
  - One key of the pair is kept private
  - The other key is made public, by uploading it to a key server, publishing it via a directory, or having a certification authority sign it into a certificate
- **To send private data, encrypt the data in the recipient's public key**
  - Only the recipient, with the corresponding private key, can decrypt it
- **To sign data, encrypt the data in your private key**
  - If the data decrypts with the corresponding public key, retrieved from a directory or key server, it came from you
  - *This is what we're interested in*



private key



public key

In DNSSEC, Zones Have Key Pairs

- **A signed zone has (at least) one key pair associated with it**
  - The private key is used to sign zone data
  - Remote recursive name servers look up the zone's public key to validate signed data in the zone

DNSSEC Record Types

- *RRSIG* Stores a Resource Record's digital SIGnature
- *DNSKEY* Stores a DNS zone's public KEY
- *DS* Allows a parent zone to vouch for a particular public key in a subzone
- *NSEC/NSEC3* Used to send validate-able negative responses (not covered)

RRSIG Records

Infoblox

Type | Domain name of signer

```
signed.infoblox.com.      3600   IN     RRSIG  DNSKEY 5 3 3600 20080410205926 20080311205926 16366 signed.infoblox.com.
      hwIdf2sPFwO9mIXlNhPakpfPh4lSoZkzyKO9dKIj4XjgH7Qj+N0Z94Kl Q5I8+tTgctge+n1W7/I0r62OhehotjS1PZyzidsn9cVLdBzypnoe6FJQ
      Q728xl5059arIlKx5aNXP7s9wDnBcifo9hRqiC8u+Ib/Afl4CHa2a4X4 0avwS9kQKoskkJJ5gqmi9PMwsZWtvXK9rTAt8Gw1vZb2bbAekeS3Zhh/
      UhCPalTlm5NGtbjhXKS4fzDPCFvthya0dhp9e5ZnUBvBGYrnvi5qVS1c LgiTDPZ5+0KNYFiwrpEdQ9lFfwGJvzYQY5ZaAtI8j4b/dYOcIgl9LqhR
      bqzKasBdZon0G0Ll3wk0BJ8UCQ2giFraI2jAo7hG5GXzGDaTEwjHtKs+ oUpI7ZWUCInicgx5iQS9KZ1iyxo14EwCAeqstheYAYLbCIdF7xrNYa0H
      tslqDRSvurbtTQLqGpw6eWbGJshAv8xaU3GG5sLExyWWV5fiDzvEahiK eYnsM2dLs7VtwEWcC/uJMwTQ7jIC4m68JVBUMz8guvvIH1aLvo0uyzSx
      pkLY8UTr0HQZ9/08wQj/9T9yUc/0iw4jk7b8/zv2vRrfvlAvG+JyEzUB Gy4e/DE9N+NNOnRSkOnIuIAMkVhKmvG5QN9Yx/ep1tj/SbyYuTldasQW
      Mqk33UjnDmw=
```

Signature (in base 64)

DNSKEY Records

Algorithm (5 is RSA/S

Public key (in base 64)

```
signed.infoblox.com.       3600   IN     DNSKEY 257 3 5 AwEAAZvf8cRF9flQim+x3vFqbKMq2uBAI2g79UApMupGNnpMncHKbzYg
       C4mn7n8GZU6QNXyWaep7g2wXQJatV4xS8JKUxXMm0S3+0mXVKPgU4otL JTSdPt+RQPxEtWLcmtT0v648OTypu2VNx7NhphBt91iQxwsI9
       60bXHTG mgiIQjBCy9wI6SD3Ay4dsZauDO/y/JJowlN+fbKnwXW7FM1bIaz/Iubg d/JKC7ofbI4jqSzrLtyIMDWYHV8xPVI6hsIPUda4Z
       SxXVAN4DZilKSN6 YcEfOaRZ1aG+g1QtsWpduspM1zXTB2srVvZkLf1TCq1g7g48Hn742QWu sxaafZT+x5kxOEKhZ4Zt/6mUue3EOgKDG
       Wu+5tbwc+VdjDW95hfPcvDn 1UD2MHq4dD3dUgWngy5paXM1PTsy8geXVQUcA5iJ5dFAjsL0oduXKs8D RneMDIUYOMlnbzHO3gv0v+QBR
       z6XPsEUcifq8kwzwO1LVg3HMSliOKYq lx9E8KWt0QOCr1lSsceNoxlli1f6sckFzNTZKpqfEa4zobtSBkdeokHv aMlAxaTuW1lqT7gVU
       vVJ8k7EnFalh6yHJ8ObfFyrKHwspkAQg19NFzi2 kpM8uGGkPsVb4ijJYK7kuyRenLWUR0ySuiMd44xj9W2TABwT7MMtZ1n6xqZC+kT9+U
       PP79+9
```

DS Records

Re

Hash algorithm (2 is SHA-25

Hash

```
signed.infoblox.com.      3600   IN     DS     12892 5 2 F1E184C0E1D615D20EB3C223ACED3B03C773DD952D5F0EB5C777586D E18DA6B5
```

More Keys Than a High School Janitor

Infoblox

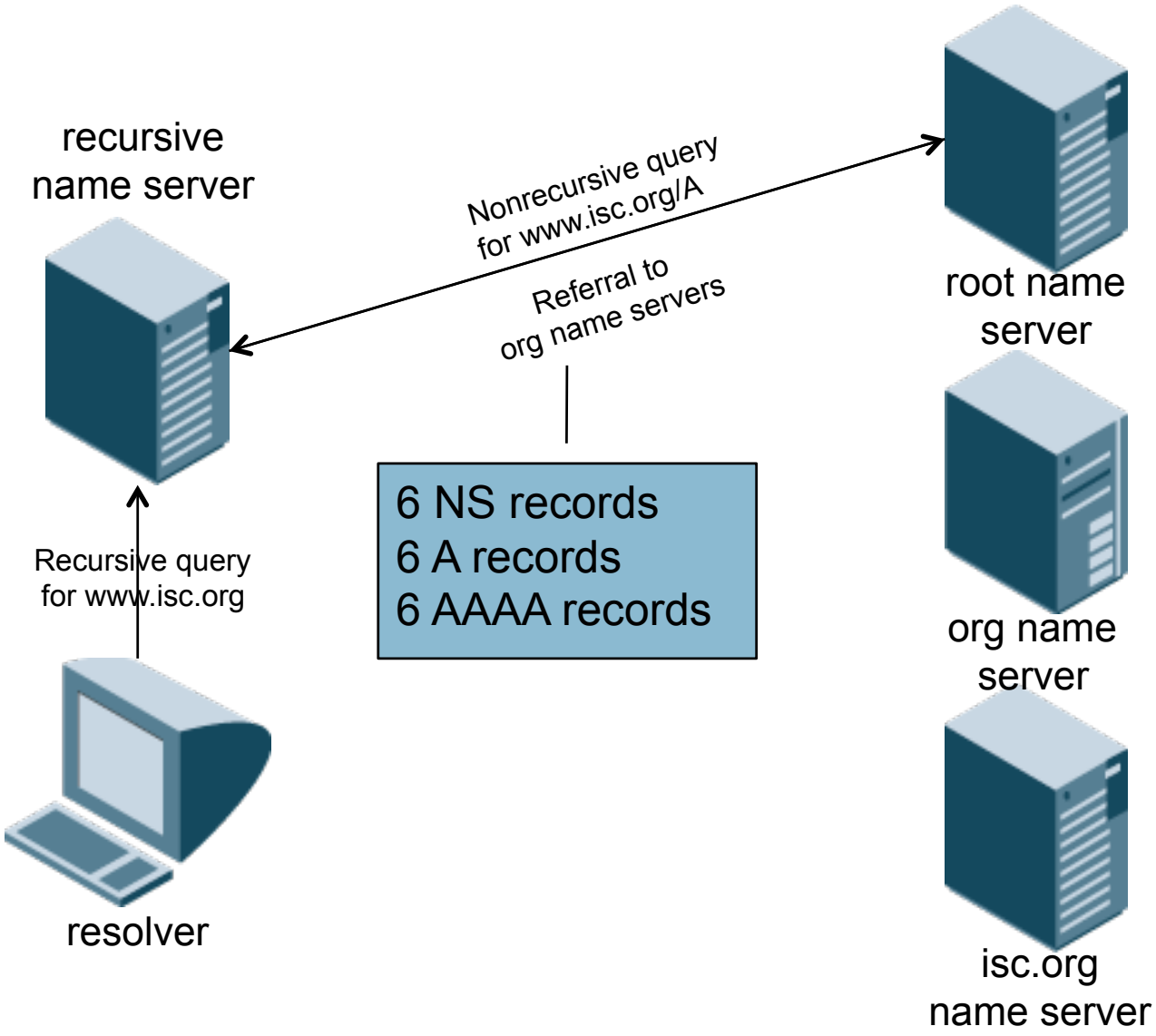
- **Most zones actually have more than one key pair**
- **After a private key has been used a lot, you're supposed to *roll over* to a new key pair**
  - All the encrypted material can be used to conduct a cryptanalysis attack
  - NIST recommends rolling over once per month
- **If your key (or a hash of it) is published in your parent zone, this is a hassle**
  - You need to communicate with your parent
  - You can't just change keys or you'll invalidate cached data
- **Consequently, we use two key pairs per zone**
  - A *zone-signing key pair, or ZSK,* used to sign all the zone data
  - A *key-signing key pair, or KSK,* used to sign just the zone's public keys
- **The ZSK is rolled over monthly, but this doesn't require notification of your parent, just re-signing with your KSK**
- **The KSK is rolled over yearly**

What Took You So Long?

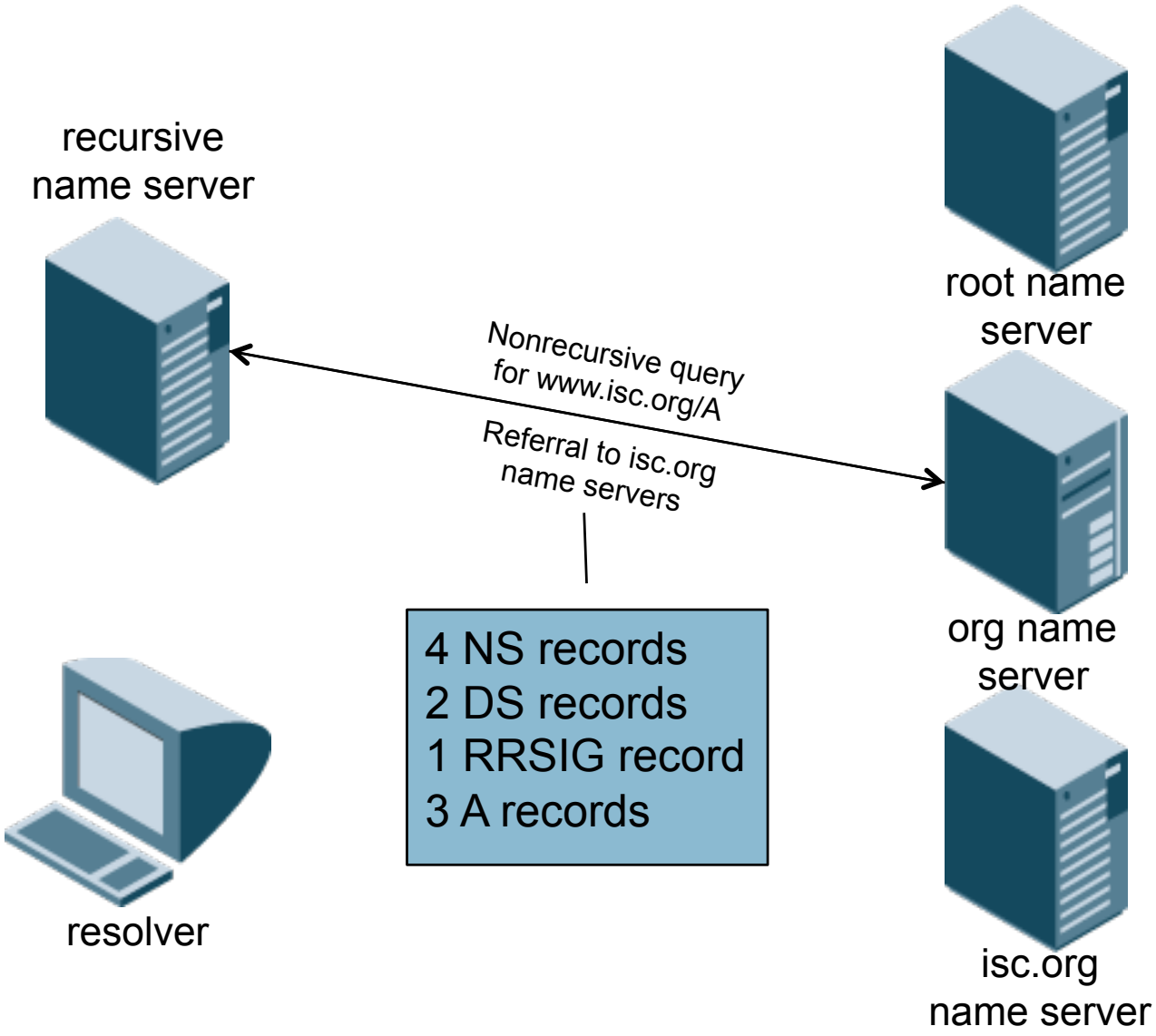
Infoblox

- **Adoption of DNSSEC has been very slow**
  - Administering signed zones is cumbersome
  - Signing makes zones much (4x-7x) bigger
  - Validating signatures requires more resources on recursive name servers

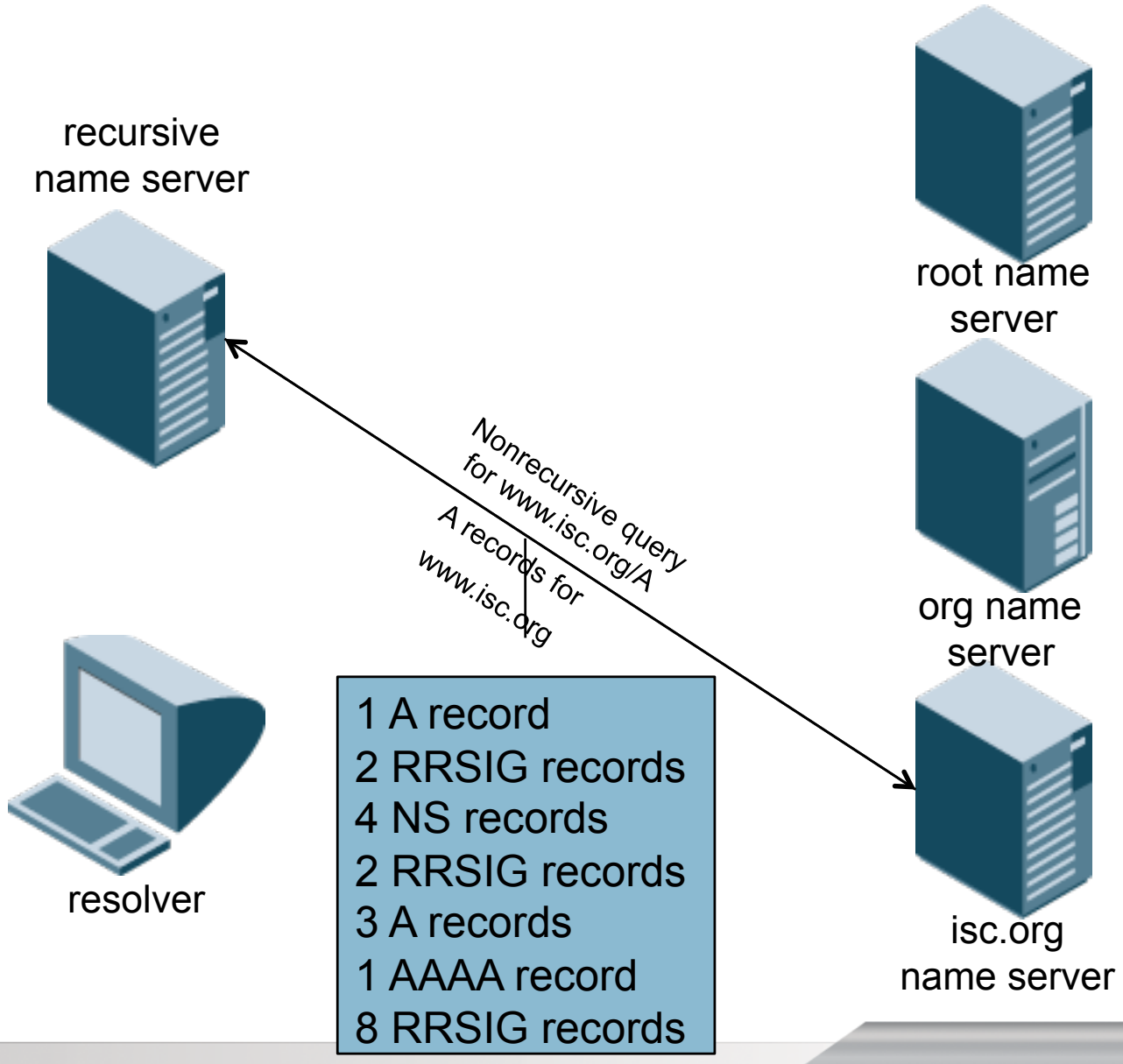
Resolution with DNSSEC



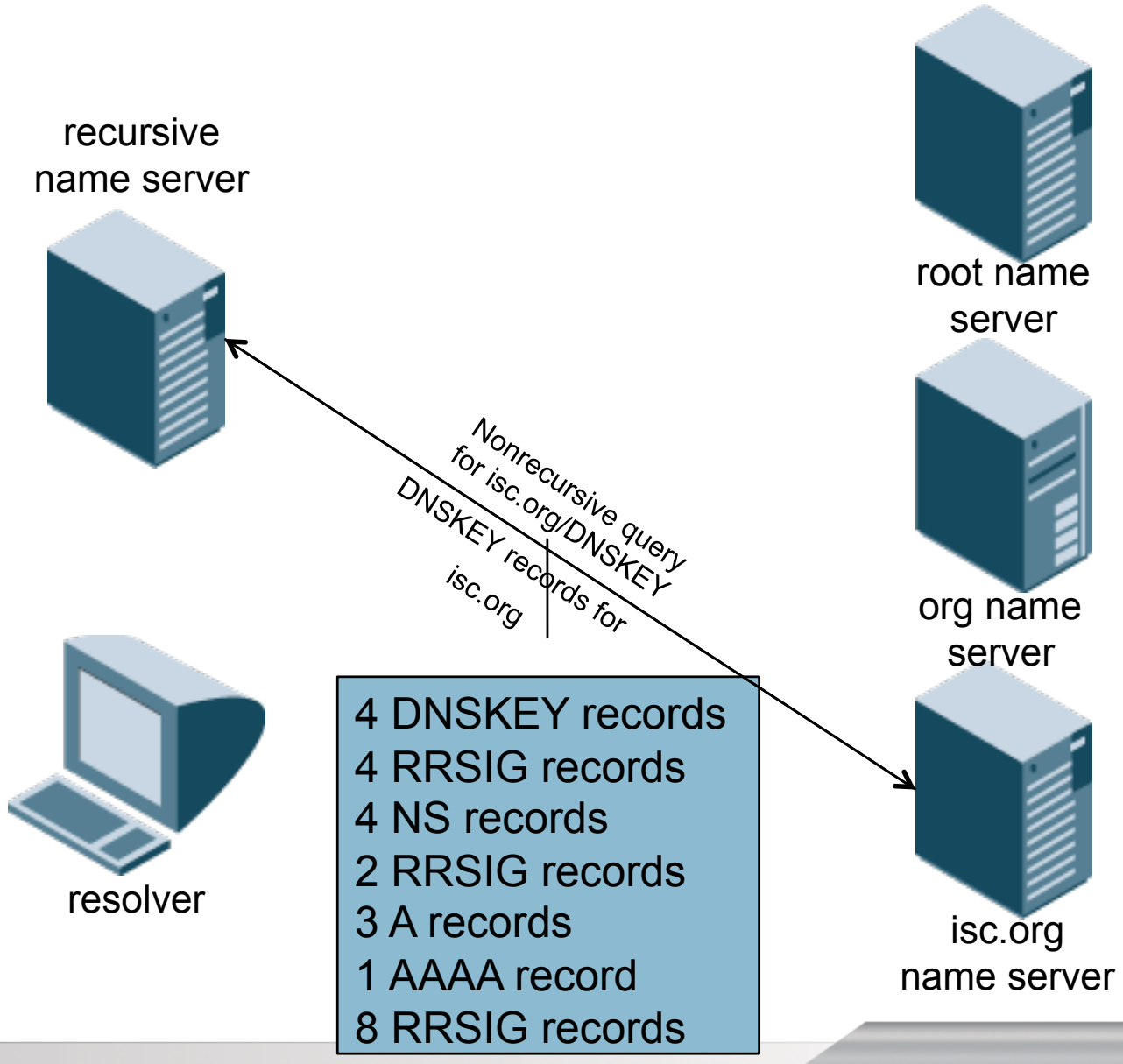
Resolution with DNSSEC



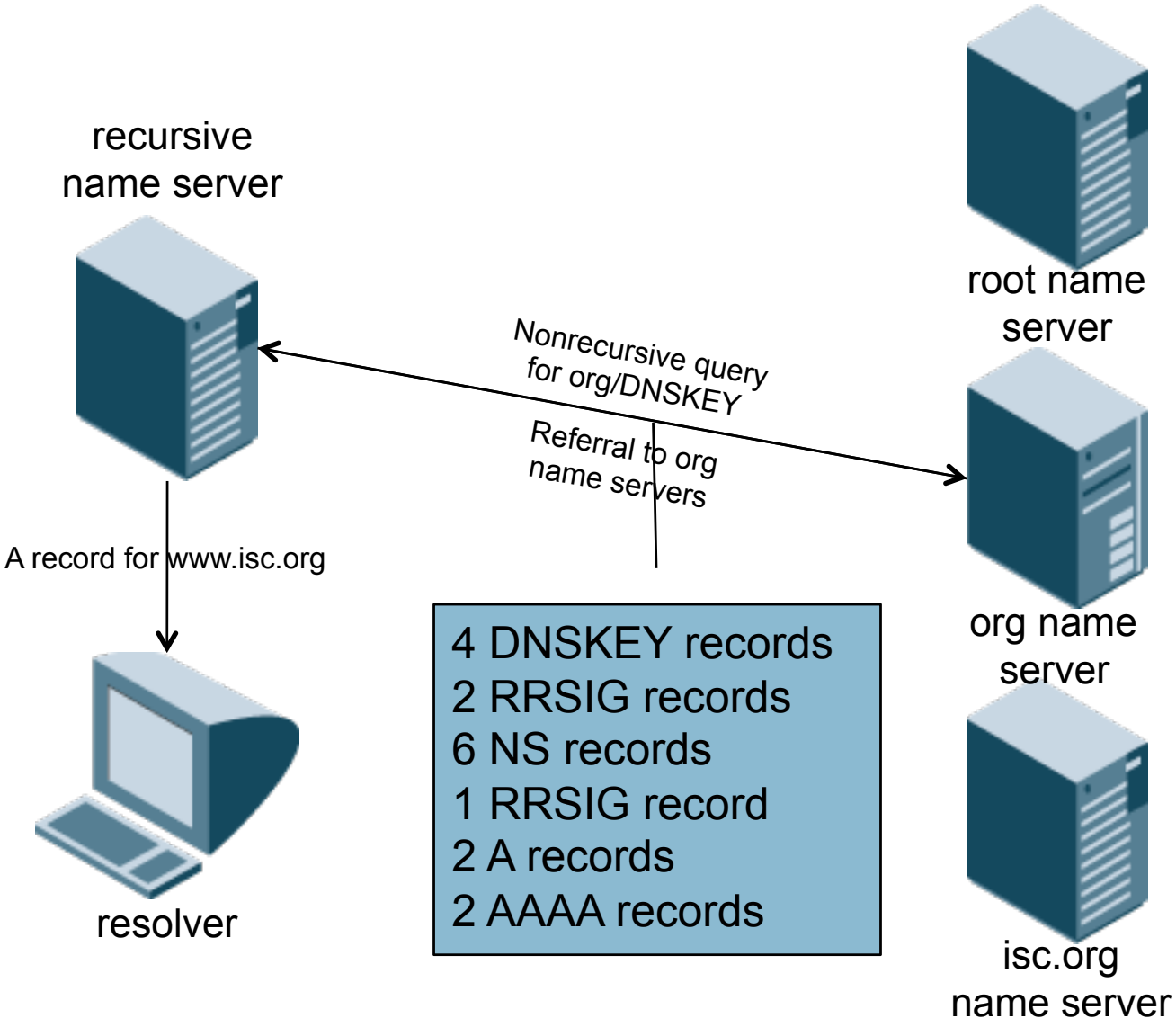
Resolution with DNSSEC



Resolution with DNSSEC



Resolution with DNSSEC



Analysis: Traffic

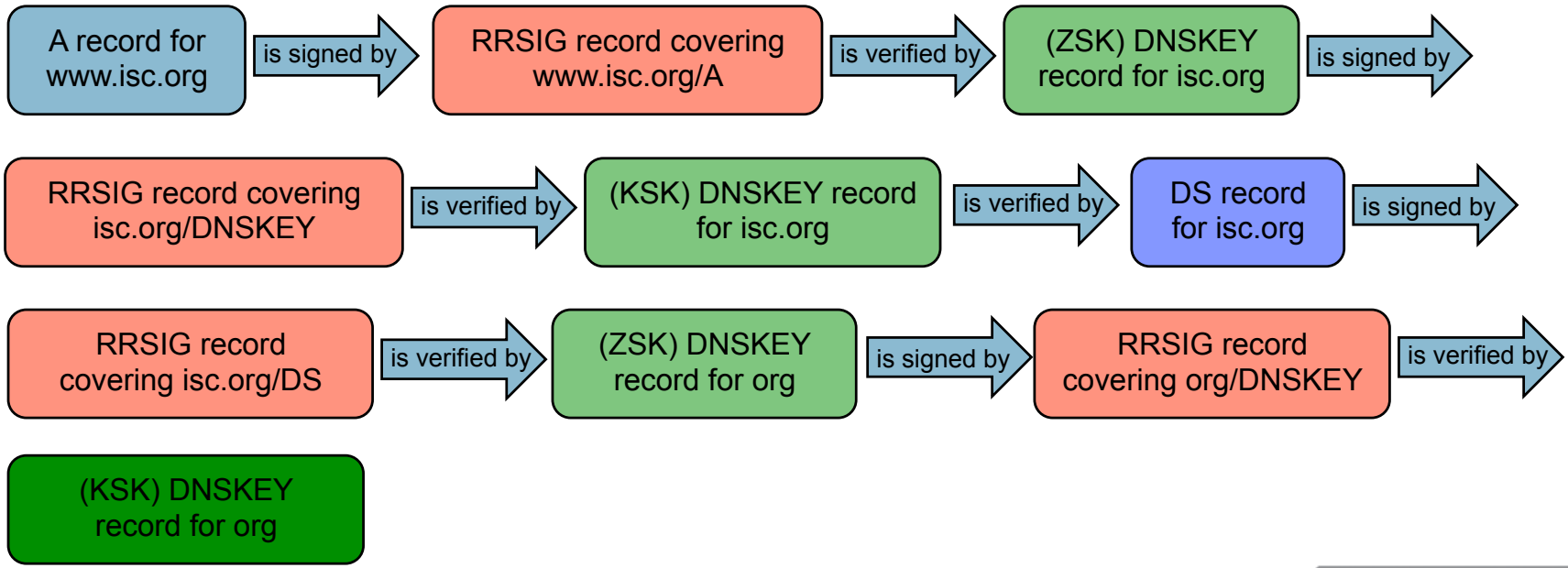
Infoblox

	www.cert.org	www.isc.org
Query 1	49 bytes	48 bytes
Response 1	451	450
Query 2	49	48
Response 2	602	442
Query 3	49	48
Response 3	133	2243
Query 4		44
Response 4		3665
Query 5		40
Response 5		1731
Total	1333 bytes	8759 bytes

That's 6.6x as much traffic!

DNSSEC Validation

- **In DNSSEC validation, a recursive name server verifies all of the signatures from the answer back to the closest *trust anchor* (a public key it knows and trusts)**
  - When DNSSEC is fully deployed, the only trust anchor necessary will be the root's public key
- **Validation can require lots of steps**
- **For example:**



Analysis: Records

Infoblox

Records Type	Count
RRSIG	20
DS	2

In order to completely validate all RRSIG and DS records (worst case):

- 22 cryptographic hashing operations
- 20 asymmetric decryption operations

Example: Rolling Over a ZSK on a Microsoft DNS Server

Performing pre-published ZSK rollover

In this procedure, ZSK1 and KSK1 denote the keys that are currently used to sign the zone. ZSK2 and KSK2 denote the new keys that will be generated using this procedure. All signing operations must continue to use KSK1 to sign the records at the apex in addition to the appropriate ZSK.

The following table provides a list of step to use when performing a pre-published ZSK rollover.

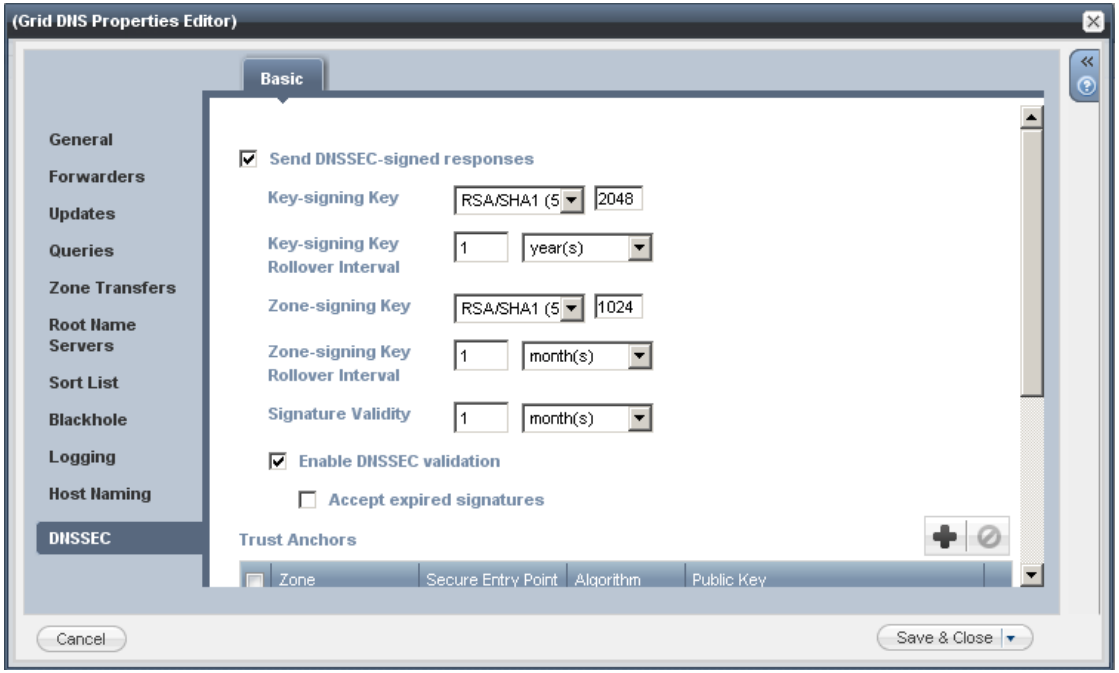
Step	Description	Command
Step 0	The zone has been signed with KSK1 and ZSK1. For more information, see Sign a Zone File.	The zone has been signed with a specified validity period using **/ValidFrom** and **/ValidTo** parameters.
Step 1	Generate the new key, ZSK2. For more information, see Generate Key Pairs.	**DnsCmd /OfflineSign /GenKey**

Example: Rolling Over a ZSK on a Microsoft DNS Server (continued)

Step 2	Identify the TTL value of the DNSKEY RRset in the zone (DNSKEY_TTL) and the maximum zone TTL (MaxZone_TTL).	
Step 3	Add the new key to the zone and re-sign the zone with KSK1 and ZSK1, using /addkey with ZSK2. Note that ZSK2 is not used to sign the zone. For an example, see Zone signing commands.	**DnsCmd /OfflineSign /SignZone** Use **/signkey** twice, once with ZSK1 and once with KSK1. Use **/addkey** to add ZSK2 to the zone.
Step 4	After re-signing, wait for a period of time equal to the MaxZone_TTL value.	
Step 5	After the period of time specified in MaxZone_TTL has elapsed, re-sign the zone with ZSK2, using /addkey with ZSK1. Note that ZSK1 is not used to sign the zone. For an example, see Zone signing commands.	**DnsCmd /OfflineSign /SignZone** Use **/signkey** to sign the zone with ZSK2. Use **/addkey** to add ZSK1 to the zone.
Step 6	Wait for a period of time equal to the MaxZone_TTL value.	
Step 7	After the period of time specified in MaxZone_TTL has elapsed, re-sign the zone with KSK1 and ZKS2. This will delete ZSK1 from the zone. For an example, see Zone signing commands.	**DnsCmd /OfflineSign /SignZone** Use **/signkey** twice with ZSK2 and KSK1. Use **/ValidFrom** and **/ValidTo** parameters to specify the validity period for ZSK2.

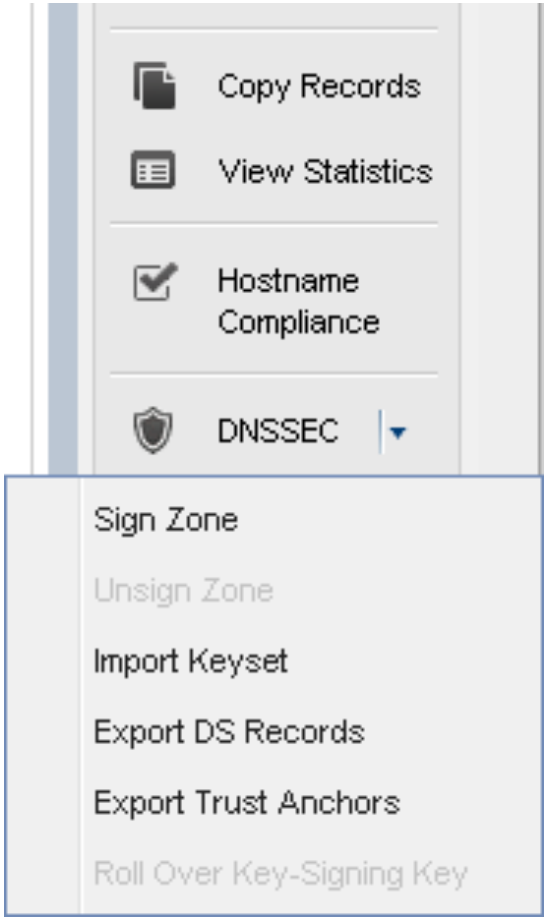
DNSSEC on Infoblox

- **Configure all DNSSEC parameters graphically, in one place**
- **Defaults according to NIST 800-81**
- **Supports NSEC3**



Maintenance of a DNSSEC-Signed Zone

- **Signing a zone**
  - Click “Sign Zone”
- **Re-signing a zone (after modifying zone data)**
  - [Automatic]
- **Rolling over a Zone-Signing Key**
  - [Automatic]
- **Configuring trust anchors for signed zones managed by the grid**
  - [Automatic]



DNSSEC Accelerating: Zero to 60? Yes!

Infoblox

- **Several European top-level zones are signed**
  - .bg, .ch, .cz, .li, .pt, .se, .uk
- **Several other top-level zones are signed**
  - .br, .na, .nu, .pr, .th, .tm
- **A few gTLDs are signed**
  - .arpa, .gov, .org
- **OMB mandated that Federal government agencies sign Internet-facing zones by the end of 2009**
  - About 20% did
- **VeriSign announced it will sign .net in 2010 and .com in 2011**
- **NTIA/DoC announced that they will sign the root zone by July 1, 2010!**

What Do I Do?

1. Upgrade to the latest version of your name server
2. Disable recursion where possible
3. Limit recursion where necessary
4. Start learning about DNSSEC

How Are We Doing?

Infoblox

- **Percentage of Internet name servers that are open recursors**

80%

- **Percentage of open recursors that are *not* patched against the Kaminsky vulnerability**

10%

- **Estimated number of Internet name servers trivially vulnerable**

1.3 million

How Are We Doing (The DNSSEC Edition)

Infoblox

- **Percentage growth in the number of DNSSEC-signed zones between the 2008 and 2009 DNS Surveys**

371%

- **Percentage of zones signed (in random sample of approximately 3.3M zones)**

.005% (167 signed zones)

Summary

Infoblox

- **All is not lost**
  - Securing DNS isn't all that hard
- **Recursion is important**
  - To spoofing
  - To denial of service attacks
- **We can help**
  - Talk to us here
  - Visit our web site for white papers, presentations and other resources

References

- “BIND 9 DNS Cache Poisoning,” Amit Klein, Trusteer, http://www.trusteer.com/docs/bind9dns.html
- “Birthday Paradox,” Wikipedia, http://en.wikipedia.org/wiki/Birthday_paradox
- 2005-2009 DNS Surveys by The Measurement Factory, http://dns.measurement-factory.com/surveys/
- Infoblox’s DNS Advisor, http://www.dnsadvisor.com

Social Media

Infoblox

- **The Ask Mr. DNS Podcast: www.ask-mrdns.com**
  - Also free on the iTunes Store
- **My blog: www.cricketondns.com**
- **On Twitter: @cricketliu, @cricketondns**