DNSSEC in your workflow
• Overview of problem space
  – Architectural changes to allow for DNSSEC deployment

• Deployment tasks
  – Key maintenance
  – DNS server infrastructure
  – Providing secure delegations
Generic view

In: registrations
Out: zone

Registry Backoffice

Registrars & Registrants

Secondary DNS

primary DNS

NLnet Labs

© 2006-2012 NLnet Labs. Licensed under a Creative Commons Attribution 3.0 Unported License.
The Registries Core business

- Maintain who is the authoritative user of the domain name
- Maintain the relation between the domain name and a number of technical parameters:
  - NS, A and AAAA
- Publish those relations in the DNS

People trust the DNS because you do a good job at this.
Introducing DNSSEC

Registrars & Registrants

DNSSEC aware Provisioning

In: registrations
Out: zone

Registry Backoffice

DNSEC Zone Signing

DNSSEC aware

DNSSEC aware

DNSSEC aware

DNSSEC aware Secondary DNS

DNSSEC aware primary DNS

Secondary DNS
DNSSEC deployment tasks

- Key maintenance policies and tools
  - Private Key use and protection
  - Public key distribution

- Zone signing and integration into the provisioning chain

- DNS server infrastructure

- Secure delegation registry changes
  - Interfacing with customers
• Overview of problem space
  • Architectural changes to allow for DNSSEC deployment

• Deployment tasks
  – Key maintenance
  • DNS server infrastructure
  • Providing secure delegations
Key Maintenance

- DNSSEC is based on public key cryptography
  - Data is signed using a private key
  - It is validated using a public key

- Operational problems:
  - Dissemination of the public key
  - Private key has a ‘best before’ date
    - Keys change, and the change has to disseminate
Public Key Dissemination

• In theory only one trust-anchor needed that of the root
  • How does the root key get to the end user?
  • How is it rolled?
• In absence of hierarchy there will be many trust-anchors
  • How do these get to the end-users?
  • How are these rolled?
• These are open questions, making early deployment difficult.
Public Key Dissemination at RIPE NCC

- In absence of a signed parent zone and automatic rollover:
  - Trust anchors are published on an “HTTPS” secured website
  - Trust anchors are signed with the RIPE NCC public keys
  - Trust anchor will be rolled twice a year (during early deployment)
  - Announcements and publications are always signed by x.509 or PGP
Key Management

• There are many keys to maintain
  • Keys are used on a per zone basis
    • Key Signing Keys and Zone Signing Keys
  • During key rollovers there are multiple keys
    • In order to maintain consistency with cached DNS data [RFC4641]
• Private keys need shielding
Approaches

• Use of a smart card to store the KSK
• The use of hardware signers and management software
  • Steep learning curve, write your own interfaces
  • https://www.centr.org/docs/2007/05/Tech16_9_Dickinson.pdf
  • http://www.nlnetlabs.nl/publications/hsm/index.html
Example implementation

• Based on Net::DNS::SEC frontend to the BIND dnssec tools
Maintaining Keys and

• The KeyDB maintains the private keys
  • It ‘knows’ rollover scenarios
  • UI that can create, delete, roll keys without access to the key material
• Physically secured
• The signer ties the Key DB to a zone
  • Inserts the appropriate DNSKEYs
  • Signs the the zone with appropriate keys
• Strong authentication
Private Key Maintenance
The software

- Perl front-end to the BIND dnssec-signzone and dnssec-keygen tools
- Key pairs are kept on disc in the “BIND format”
- Attribute files containing human readable information
  - One can always bail out and sign by hand.

- Works in the RIPE NCC environment, is a little rough edged but available via the www.ripe.net/disi
Example session

$ maintkeydb create KSK RSASHA1 2048 example.net
  Created 1 key for example.net
$ maintkeydb create ZSK RSASHA1 1024 example.net
  Created 2 keys for example.net
$ dnssigner example.net
  Output written to :example.net.signed

$ maintkeydb rollover zsk-stage1 RSASHA1 example.net
OpenDNSSEC

• A framework to maintain your signed zones.
• All based on one-off configuration
• Work towards a true bump in the wire
  • Enforcer NG (expected in v2.0, July)
  • Signer NG input output modules in v1.4 (now in alpha)
• www.opendnssec.org for more information
Presentation roadmap

- Overview of problem space
  - Architectural changes to allow for DNSSEC deployment

- Deployment tasks
  - Key maintenance
    - DNS server infrastructure
  - Providing secure delegations
Infrastructure

• One needs primary and secondary servers to be DNSSEC protocol aware

• We had a number of concerns about memory CPU and network load
  • Research done and published as RIPE 352
  • Old work; but take this as inspiration and the conclusions still hold
Conclusion from RIPE 352

- CPU, Memory and Bandwidth usage increase are not prohibitive for deployment of DNSSEC on k.root-servers.net and ns-pri.ripe.net

- Bandwidth increase is caused by many factors
  - Hard to predict but fraction of DO bits in the queries is an important factor
- CPU impact is small, Memory impact can be calculated
- Don’t add DNSKEY RR set in additional
Question

What would be the immediate and initial effect on memory, CPU and bandwidth resources if we were to deploy DNSSEC on RIPE NCC’s ‘primary’ name server?

• Measure through simulation.
The “DISTEL” Test Lab

Diagram showing the connections between the Recorder, DNS Server, 100baseT Experiment network, Queries, Answers, Player, and Control network.
DISTEL LAB

• Player plays libpcap traces in real time
  – libpcap traces are modified to have the servers
destination address
• Server has a default route to the recorder
• Recorder captures answers

• 2 Ghz Athlon based hardware with 1 Gb memory
  and 100baseT Ethernet
This Experiment

• Traces from production servers:
  – k.root-servers.net
  – ns-pri.ripe.net

• Server configured to simulate the production machines.
  – ns-pri.ripe.net
    • Loaded with all 133 zones.
  – k.root-servers.net
    • Only loaded with the root zone.
Zone Signing

- 1 Key Signing Key 2048 bit RSASHA1
- 2 Zone Signing Keys of equal length
  - length varied between 512 and 2048
  - Only one ZSK used for signing
    - This is expected to be a common situation (Pre-publish KSK rollover)

- 3 DNSKEY RRs in per zone
  - 1 RRSIG per RR set
  - 2 RRSIGs over the DNSKEY RR set
Loading the Zones: Memory Use

• Various zone configurations were loaded.
  – Mixtures of signed and unsigned zones
  – Memory load for different numbers of RRSIGs and NSECs.

• Memory load is implementation and OS specific
Memory

- On ns-pri.ripe.net factor 4 increase.
  - From ca. 30MB to 150MB
  - No problem for a 1GB of memory machine
- On k.root-servers.net
  - Increase by ca 150KB
  - Total footprint 4.4 MB
- Nothing to worry about
- Memory consumption on authoritative servers can be calculated in advance.
  - No surprises necessary
Serving the zones
Query Properties

• DNS clients set the “DO” flag and request for DNSSEC data.
  − Not to do their own validation but to cache the DNSSEC data for.

• EDNS size determines maximum packet size.
  (DNSSEC requires EDNS)

• EDNS/DO properties determine which fraction of the replies contain DNSSEC information
EDNS properties

k.root DNS Packets

k.root EDNS size Distribution

EDNS size for "DO" queries in k.root

ns-pri DNS Packets

ns-pri EDNS Size Distribution

EDNS size for "DO" queries in ns-pri
Serving the zones

• Measured for different keysizes.
  – named for ns-pri.ripe.net
  – nsd and named for ns-pri.ripe.net and k.root-servers.net

• We also wanted to study “worst case”;
  What if all queries would have the DO bit set?
  – Modified the servers to think that queries had EDNS 2048 octets size and DO bit set
<table>
<thead>
<tr>
<th>Trace Server</th>
<th>ZSK Size</th>
<th>WCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns-pri BIND 9.3.1</td>
<td>0000</td>
<td>ca 14%</td>
</tr>
<tr>
<td>ns-pri BIND 9.3.1</td>
<td>2048</td>
<td>ca 18%</td>
</tr>
<tr>
<td>k.root BIND 9.3.1</td>
<td>0000</td>
<td>ca 38%</td>
</tr>
<tr>
<td>k.root BIND 9.3.1</td>
<td>2048</td>
<td>ca 42%</td>
</tr>
<tr>
<td>k.root BIND 9.3.1 mod</td>
<td>2048</td>
<td>ca 50%</td>
</tr>
<tr>
<td>k.root NSD 2.3.0</td>
<td>0000</td>
<td>ca 4%</td>
</tr>
<tr>
<td>k.root NSD 2.3.0</td>
<td>2048</td>
<td>ca 4%</td>
</tr>
<tr>
<td>k.root NSD 2.3.0 mod</td>
<td>2048</td>
<td>ca 5%</td>
</tr>
</tbody>
</table>
Bandwidth Factors

• fraction of queries with DO bit
  – Seen in difference between ns-pri and k.root result
  – Seen in difference between modified and unmodified servers

• Including DNSKEY RR in additional section.
  – Seen in difference between k.root traces from modified nsd and modified named

• Difference in answer patterns
  – Name Errors vs Positive answers
  – Difficult to assess from this data
Trace k.root against modified nsd 2.3.0

Bandwidth Increase

Upper Bound
Upper Bound
Bandwidth observation

• DNSKEY RR set with RRSIG in the additional section
  – Fairly big chunk of data
  – None of the clients today validate the data
  – Clients that need the data will query for it

• Servers MAY include the DNSKEY Rrset

• NSD does not include

• Named does include
  – Recommendation to make the inclusion configurable
Bandwidth Increase

• Significant for ns-pri.ripe.net
  – Well within provisioned specs.
• Insignificant for k.root-servers.net
  – Upper bound well within provisioning specs
    • even when including DNSKEY RR set in additional section

(Key size influences bandwidth but bandwidth should not influence your key size)
Not Measured

• The experiment has been done in a closed environment

• What about the behavior of clients that do expect DNSSEC information but do not receive it?
  – Firewalls dropping packets with DNSSEC
  – BIND behavior is well understood

• What about implementations that set the DO bit but cannot handle DNSSEC data that is returned?

• Measure these on the Internet
Conclusion of these measurements

• CPU, Memory and Bandwidth usage increase are not prohibitive for deployment of DNSSEC on k.root-servers.net and ns-pri.ripe.net

• Bandwidth increase is caused by many factors
  – Hard to predict but fraction of DO bits in the queries is an important factor

• CPU impact is small, Memory impact can be calculated

• Don’t add DNSKEY RR set in additional
More questions?

• Can you deal with TCP?
• What is the effect of NSEC3 Hashes?

• See the material on the training site
Monitoring!!

• Are you monitoring your DNSSEC Setup?
  • [http://exchange.nagios.org/directory/Plugins/Network-Protocols/DNS/check_dnssec/details](http://exchange.nagios.org/directory/Plugins/Network-Protocols/DNS/check_dnssec/details)
  • [http://secspider.cs.ucla.edu/](http://secspider.cs.ucla.edu/)
  • DNSSEXY (forthcoming)
Presentation roadmap

• Overview of problem space
  • DNSSEC in 3 slides
  • Architectural changes to allow for DNSSEC deployment

• Deployment tasks
  • Key maintenance
  • DNS server infrastructure
    – Providing secure delegations
Parent-Child Key Exchange

• In the DNS the parent signs the “Delegations Signer” RR
  – A pointer to the next key in the chain of trust

  $ORIGIN net.
  kids NS    ns1.kids
  DS  (...)  1234
  RRSIG DS  (...) net.

  money NS   ns1.money
  DS  (...)   
  RRSIG DS  (...) net.

  $ORIGIN kids.net.
  @ NS     ns1
  RRSIG NS  (...) kids.net.
  DNSKEY  (...) (1234)
  DNSKEY  (...) (3456)
  RRSIG dnskey ... 1234 kids.net. ...
  RRSIG dnskey ... 3456 kids.net. ...
  beth  A  127.0.10.1
  RRSIG A  (...) 3456 kids.net. ...

• DNSKEY or DS RR needs to be exchanged between parent and child
Underlying Ideas

• The DS exchange is the same process as the NS exchange
  • Same authentication/authorization model
  • Same vulnerabilities
  • More sensitive to mistakes
• Integrate the key exchange into existing interfaces
  • Customers are used to those
• Include checks on configuration errors
  • DNSSEC is picky
• Provide tools
  • To prevent errors and guide customers
Changes your core business?

- Maintain who is the authoritative user of the domain name
- Maintain the relation between the domain name and a number of technical parameters:
  - NS, A, AAAA and DS
- Publish those relations in the DNS

*Users can now be confident that they get the data as published by the party they trust*
Recursive Name server

• Resource needs
  • Early deployment: little actual crypto
  • See graph next slide
  • Early deployment: some bugs enhanced troubleshooting
• Trust Anchor Maintenance
  • A new responsibility!
Questions and Discussion