The details

Presented by

Olaf Kolkman (NLnet Labs)
DNSSEC Mechanisms

- New Resource Records
- Setting Up a Secure Zone
- Delegating Signing Authority
Data flow through the DNS
Where are the vulnerable points?

- Registrars & Registrants
- Secondary DNS
- Primary DNS
- Registry
- Server vulnerability
- Man in the Middle
- Man in the Middle
- Spoofing & Man in the Middle

Server vulnerability

© 2006-2012 NLnet Labs, Licensed under a Creative Commons Attribution 3.0 Unported License.
DNSSEC protects all these end-to-end

• As an aside:
  There is a protection mechanism against the man in the middle: TSIG
  • Provides hop-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 confidentiality
- The seal protects the delivery process
- No assertion about the message
Data flow through the DNS
End to end security
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
The mechanism used

• Using public key cryptographic algorithms, signatures are applied over the DNS data.

• By comparing the signatures with public keys, the integrity and authenticity of the data can be established.
Public key cryptography in a nutshell

- Two large numbers and an encryption and decryption algorithm
- If one of the numbers (the private key) and a message are used for encryption
- The other number (public key) and the decryption algorithm can be used to retrieve the original message
Message

Private Key

Encrypt

Public Key

Message

Decryption only with matching key: If you can decrypt with a public key you may assert the message was signed with corresponding private key.
Use that for signatures

Message

Message Digest

Private Key

Signature

Public Key

Decrypted Message Digest

Calculated Message Digest

plain text

Message

© 2006-2012 NLnet Labs, Licensed under a Creative Commons Attribution 3.0 Unported License.
In Practice

- Key generation and signing is done by tools
- Validating and signing entity need to communicate which algorithms for hashing and public key cryptography is needed: e.g. RSASHA1, RSASHA256 or DSA
Holy Trinity

- Private Key: kept private and stored locally
- Public Keys: Published in the DNS as a DNSKEY Resource Record
- Signatures: Published in the DNS as a RRSIG Resource Record
Signing is done per Zone

• Each zone has one or more key-pairs for signing

• If you have the public keys from a zone you can validate signatures made with the corresponding private keys

• However, signing a complete zone does not scale
### RRs and RRSets

- **Resource Record:**
  - name    TTL   class   type   rdata
  - `www.nlnetlabs.nl.  7200  IN  A  192.168.10.3`

- **RRset: RRs with same name, class and type:**
  - `www.nlnetlabs.nl.  7200  IN  A  192.168.10.3`
  - `A  10.0.0.3`
  - `A  172.25.215.2`

- **RRsets are the atomic data units in the DNS**
- **RRsets are signed, not the individual RRs**
DNSKEY RDATA

- 16 bits: FLAGS
- 8 bits: protocol
- 8 bits: algorithm
- N*32 bits: public key

nlnetlabs.nl. 3600 IN DNSKEY 256 3 5 (AQOvhvXXU61Pr8sCwELcqqq1g4JJCALG4C9EtraBKVd+vGIF/unwigfLOA03nHp/cgGrG6gJYe8OWKYNgq3kDChN)
RRSIG RDATA

• 16 bits - type covered
• 8 bits - algorithm
• 8 bits - nr. labels covered
• 32 bits - original TTL

nlnetlabs.nl.  3600 IN  RRSIG  A  5  2  3600  (  
20050611144523 20050511144523 3112 nlnetlabs.nl.  
VJ+8ijXvbrTLeoAiEk/qMrdudRnYZM1V1qhN  
vhYuAcYKe2X/jqYfMfjfSUrhmPo+0/GOZjW  
66DJubZPmNSYXw==  )

• 32 bit - signature expiration
• 32 bit - signature inception
• 16 bit - key tag
• signer’s name
Validate Public Keys

- Make sure you get them from the appropriate entity and configure them as trust-anchors
- If you validate against the wrong public key there is a problem again
- For DNSSEC: key distribution through the DNS
  - Ideally only one key needed: that of the root of the DNS hierarchy (more on that later)
Delegating Signing Authority

Chains of Trust
• Key distribution does not scale!

Secure entry points

Out of band key-exchanges
Locally Secured Zones

- Delegate Signing Security

**NS and DS**

```
+-----------------+            +-----------------+            +-----------------+
|                 |            |                 |            |                 |
| NS & DS         |            | NS & DS         |            | NS & DS         |
|                 |            |                 |            |                 |
| net.            |            | com.            |            |                 |
|                 |            |                 |            |                 |
| money.net.      |            | kids.net.       |            | os.net.         |
|                 |            |                 |            |                 |
| corp            |            | dop             |            | mac             |
|                 |            |                 |            | unix            |
| dev             |            | marnick         |            | nt              |
| market          |            | dilbert         |            |                 |
```

Secure entry points

Wednesday, February 15, 12
Using the DNS to Distribute Keys

• Secured islands make key distribution problematic

• Distributing keys through DNS:
  – Use one trusted key to establish authenticity of other keys
  – Building chains of trust from the root down
  – Parents need to sign the keys of their children

• Only the root key needed in ideal world
  – Parents always delegate security to child
Delegation Signer (DS)

• Delegation Signer (DS) RR indicates that:
  – delegated zone is digitally signed
  – indicated key is used for the delegated zone

• Parent is authoritative for the DS of the child’s zone
  – Not for the NS record delegating the child’s zone!
  – DS **should not** be in the child’s zone
• 16 bits: key tag
• 8 bits: algorithm
• 8 bits: digest type
• 20 bytes: SHA-1 Digest

$ORIGIN nlnetlabs.nl.
lab.nlnetlabs.nl.  3600 IN   NS  ns.lab.nlnetlabs.nl
lab.nlnetlabs.nl.  3600 IN   DS  3112   5   1  (  
239af98b923c023371b52  
1g23b92da12f42162b1a9  )
Key Problem

- Interaction with parent administratively expensive
  - Should only be done when needed
  - You might want to lock these in hardware
- Signing zones should be fast
  - Memory restrictions
  - Space and time concerns
  - Operational exposure higher
More Than One Key: KSK and ZSK

- RRsets are signed, not RRs
- DS points to specific key
  - Signature from that key over DNSKEY RRset transfers trust to all keys in DNSKEY RRset
- Key that DS points to only signs DNSKEY RRset
  - Key Signing Key (KSK)
- Other keys in DNSKEY RRset sign entire zone
  - Zone Signing Key (ZSK)
The Important Considerations

- KSK and ZSK have different ‘shielding’ properties: KSK on smartcard, ZSK on disk
- ZSK needs ‘daily’ or permanent use.
- KSK less frequent
- ZSK change needs no involvement with 3rd parties
- KSK may need uncontrolled cooperation from 3rd parties
Initial Key Exchange

- Child needs to:
  - Send key signing keyset to parent

- Parent needs to:
  - Check child's zone
    - for DNSKEY & RRSIGs
  - Verify if key can be trusted
  - Generate DS RR
Locally configured
Trusted key: . 8907
$ORIGIN .

DNSKEY (…) 5TQ3s… (8907) ; KSK
DNSKEY (…) lasE5… (2983) ; ZSK

RRSIG DNSKEY (…) 8907 . 69Hw9..
net. DS 7834 3 1ab15…
RRSIG DS (…) 2983

$ORIGIN net.

net. DNSKEY (…) q3dEw… (7834) ; KSK
DNSKEY (…) 5TQ3s… (5612) ; ZSK
RRSIG DNSKEY (…) 7834 net. cMas...

$ORIGIN foo.net.

foo.net. DNSKEY (…) rwx002… (4252) ; KSK
DNSKEY (…) sovP42… (1111) ; ZSK
RRSIG DNSKEY (…) 4252 foo.net. 5t...

www.foo.net. A 193.0.0.202
RRSIG A (…) 1111 foo.net. a3...

Wednesday, February 15, 12
Chain of Trust Verification, Summary

- Data in zone can be trusted if signed by a Zone-Signing-Key
- Zone-Signing-Keys can be trusted if signed by a Key-Signing-Key
- Key-Signing-Key can be trusted if pointed to by trusted DS record
- DS record can be trusted
  - if signed by the parents Zone-Signing-Key
  - or
  - DS or DNSKEY records can be trusted if exchanged out-of-band and locally stored (Secure entry point)
Where are we

- DNSKEY
- RRSIG
- DS
Offline Signing and Denial of Existence

• Problems with on-the-fly signing
  • Private key needs to be stored on an Internet facing system
  • Performance, signing is a CPU expensive operation
• How does one provide a proof that the answer to a question does not exist?
• Points to the next domain name in the zone
  – also lists what are all the existing RRs for “name”
  – NSEC record for last name “wraps around” to first name in zone

• N*32 bit type bit map

• Used for authenticated denial-of-existence of data
  – authenticated non-existence of TYPEs and labels

• Example:

  www.nlnetlabs.nl. 3600 IN NSEC nlnetlabs.nl. A RRSIG NSEC
NSEC Records

• NSEC RR provides proof of non-existence
• If the servers response is Name Error (NXDOMAIN):
  – One or more NSEC RRs indicate that the name or a wildcard expansion does not exist
• If the servers response is NOERROR:
  – And empty answer section
  – The NSEC proves that the QTYPE did not exist
• More than one NSEC may be required in response
  – Wildcards
• NSEC records are generated by tools
  – Tools also order the zone

Wednesday, February 15, 12
NSEC Walk

• NSEC records allow for zone enumeration
• Providing privacy was not a requirement at the time
• Zone enumeration is a deployment barrier

• Solution has been developed: NSEC3
  • RFC 5155
  • Complicated piece of protocol work
  • Hard to troubleshoot
  • Only to be used over Delegation Centric Zones
NSEC3

- Creates a linked list of the hashed names
- Non-existence proof of the hash proofs non-existence of original
- Dictionary attack barriers:
  - Salt
  - Iterations
New Resource Records

• Three Public key crypto related RRs
  – RRSIG: Signature over RRset made using private key
  – DNSKEY: Public key, needed for verifying a RRSIG
  – DS: Delegation Signer; ‘Pointer’ for building chains of authentication

• One RR for internal consistency
  – NSEC and NSEC3: Indicates which name is the next one in the zone and which typecodes are available for the current name
    • authenticated non-existence of data
Other Keys in the DNS

• DNSKEY RR can only be used for DNSSEC
  – Keys for other applications need to use other RR types

• CERT
  – For X.509 certificates

• Application keys under discussion/development
  – IPSECKEY
  • SSHFPSummary for now

• DANE!!!
Summary and

• You have seen the new RRs and learned what is their content
Summary

• Scaling problem: secure islands

• Zone signing key, key signing key

• Chain of trust

Questions?