DNSeC Mechanisms

- New Resource Records
- Setting Up a Secure Zone
- Delegating Signing Authority
- Key Rollovers
- AD, CD, DO bits
Public Key Crypto
(in one slide)

- Key pair: a secret (or private) key and a public key
- Simplified:
  - If you know the public key, you can decrypt data encrypted with the secret key
    - Usually an encrypted hash value over a published piece of information; the owner is the only person who can construct the secret. Hence this a signature
  - If you know the secret key, you can decrypt data encrypted with the public key
    - Usually an encrypted key for symmetric cipher
- PGP uses both, DNSSEC only uses signatures
- Algorithms: RSA, DSA, Elliptic curve, etc...
Public Key Issues

• Public keys need to be distributed.

• Private keys need to be kept private

• Both key distribution and secrecy are not trivial

• Public key cryptography is ‘slow’
The DNS is Not a PKI

• All key procedures are based on local policy

• A PKI is as strong as its weakest link
  – Certificate Authorities control this through SLAs

• The DNS does not have Certificate Revocation Lists

• If the domain is under one administrative control you might be able to enforce policy
Security Status of Data (RFC4035)

- Secure
  - Resolver is able to build a chain of signed DNSKEY and DS RRs from a trusted security anchor to the RRset

- Insecure
  - Resolver knows that it has no chain of signed DNSKEY and DS RRs from any trusted starting point to the RRset

- Bogus
  - Resolver believes that it ought to be able to establish a chain of trust but for which it is unable to do so
  - May indicate an attack but may also indicate a configuration error or some form of data corruption

- Indeterminate
  - Resolver is not able to determine whether the RRset should be signed
New Resource Records
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 signed, not the individual RRs
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 Indicates which name is the next one in the zone and which typecodes are available for the current name
  • authenticated non-existence of data
DNSKEY RDATA

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

Example:
nlnetlabs.nl. 3600 IN DNSKEY 256 3 5 (AQOvhvXXU61Pr8sCwELcqqq1g4JJ CALG4C9EtraBKVd+vGlf/unw wigfLOA O3nHp/cgGrG6gJYe80WKYNqg3kDChN)
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/qMrdudRnYZM1VlqhN  
vhYuAcYKe2X/jqYfMjfSUrmhPo+0/GOZjW  
66DJubZPmNSYXw== )

- 32 bit - signature expiration
- 32 bit - signature inception
- 16 bit - key tag
- signer’s name
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
DS RDATA

• 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

)
NSEC RDATA

• 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
NSEC Walk

• NSEC records allow for zone enumeration
• Providing privacy was not a requirement at the time
• Zone enumeration seems to be an deployment barrier

• NSEC-3 helps solved the problem
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
  – SSHFP
Delegating
Signing
Authority
Chains of Trust
Locally Secured Zones

- Key distribution does not scale!

Out of band key-exchanges

Secure entry points
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
Key Problem

• Interaction with parent administratively expensive
  – Should only be done when needed
  – Bigger keys are better

• Signing zones should be fast
  – Memory restrictions
  – Space and time concerns
  – Smaller keys with short lifetimes are better
Key solution: 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)
Initial Key Exchange

• Child needs to:
  – Send key signing keyset to parent

• Parent needs to:
  – Check childs 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...

Walking the Chain of Trust
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)
Key Rollovers
Private Keys

• You have to keep your private key secret
• Private key can be stolen
  – Put the key on stand alone machines or on bastion hosts behind firewalls and strong access control
• Private key reconstruction (crypto analysis)
  – Random number not random
  – Leakage of key material (DSA)
  – Brute force attacks
Key Rollovers

• Try to minimise impact
  – Short validity of signatures
  – Regular key rollover

• Remember: DNSKEYs do not have timestamps
  – the RRSIG over the DNSKEY has the timestamp

• Key rollover involves second party or parties:
  – State to be maintained during rollover
  – Operationally expensive
Timing of the Scheduled Key Rollover

• Don’t remove the old key while there are servers still handing out the old DS RR
• New DS needs to be distributed to the slaves
  – Max time set by the SOA expiration time
• Old DS needs to have expired from caches
  – Set by the TTL of the original DS RR
• You (or your tool) can check if the master and slave have picked up the change
Timing Properties

- **Authoritative Master**: Foo TXT Old
- **Authoritative Slave**: Foo TXT Old
- **Caching Nameserver**: Foo TXT New

- **Publication of new data**
- **Query to slave** followed by Caching
- **Zone transfer**
- **Expiration From Cache**

Time:
- 0
- $t_1$
- $t_2$
- $t_3$
Unscheduled Rollover Problems

• Needs out of band communication
  – With parent and pre-configured resolvers
• The parent needs to establish your identity again
• How to protect child delegations?
  – Unsecured?
• There will be a period that the “stolen” key can be used to generate seemingly secure data
• Emergency procedure must be on the shelf
Key Rollover - Summary

- Generate new KSK
- Sign with old and new KSKs
- Wait for your servers + TTL of old DNSKEY RRset
- Inform resolvers of the new key
  - that have you as a trusted entry point
- Query for the parental DS and remember the TTL
- Upload the new KSK or DS to the parent
- Check if *all* parental servers have new DS
- Wait another TTL before removing the old key
D0 bit

- A state bit in the « header » section of DNS packets
  - Not used before DNSSEC (should be set to zero)
  - 1 = “resolver” want DNSSEC RRs
  - 0 = “resolver” does not want DNSSEC RRs
AD bit

- A state bit in the « header » section of DNS packets
  - Not used before DNSSEC (should be set to zero)
  - Only used in response from validators
- AD bit is not set by authoritative server, unless it has been configured to do so.
- AD = Authenticated data
Bit CD

- A state bit in the « header » section of DNS packets
  - Not used before DNSSEC (should be set to zero)
- CD = Checking Disable
  - 1 = validation disable
    - “resolver” accepts non verified data
  - 0 = validation enabled
    - “resolver” want validated answers for signed data, but accepts answers for non signed data
“new” Developments

• NSEC3
  – RFC 5155
  – All RR names hashed
  – Hashed names are ordered
  – “opt-out” for unsecured delegations possibilities

• Automated Trust anchors rollover
  – RFC5011

• SHA1 to be deprecated
  – New hash for DS records
  – Overlap, no flag day
Some issues with DNSSEC

- Does not protect against denial of service attacks, but increases the risks
  - Cryptographic load
  - Larger DNSSEC messages
  - RFC5358
- Does not protect non signed RRs (non authoritative data at delegation point)
  - NS and glue in parent zone
  - Zone transfer should be protected by other means
- Add complexity to DNS, increasing the risks of bad configuration
  - Nothing is for free :-)
- How to distribute and roll trust anchor(s) ?
  - RFC5011 ?
Some issues DNSSEC (cont.d)

- NSEC offers zone-walk
  - NSEC3
- Certain firewalls/middle boxes do not support DNS message > 512 byte (edns0)
  - Many are reconfigurable
- Certain Firewalls/middle boxes have issues with AD, CD, DO bits in the DNS packet header
- Certain old stub resolvers may have issues with the AD bit
  - Add the AD bit in request for signaling resolvers state?
Questions ??