JOHANNES KEPLER UNIVERSITY LINZ | JKU

DNSSEC

Michael Sonntag

Institute for Information processing and microprocessor technology (FIM)

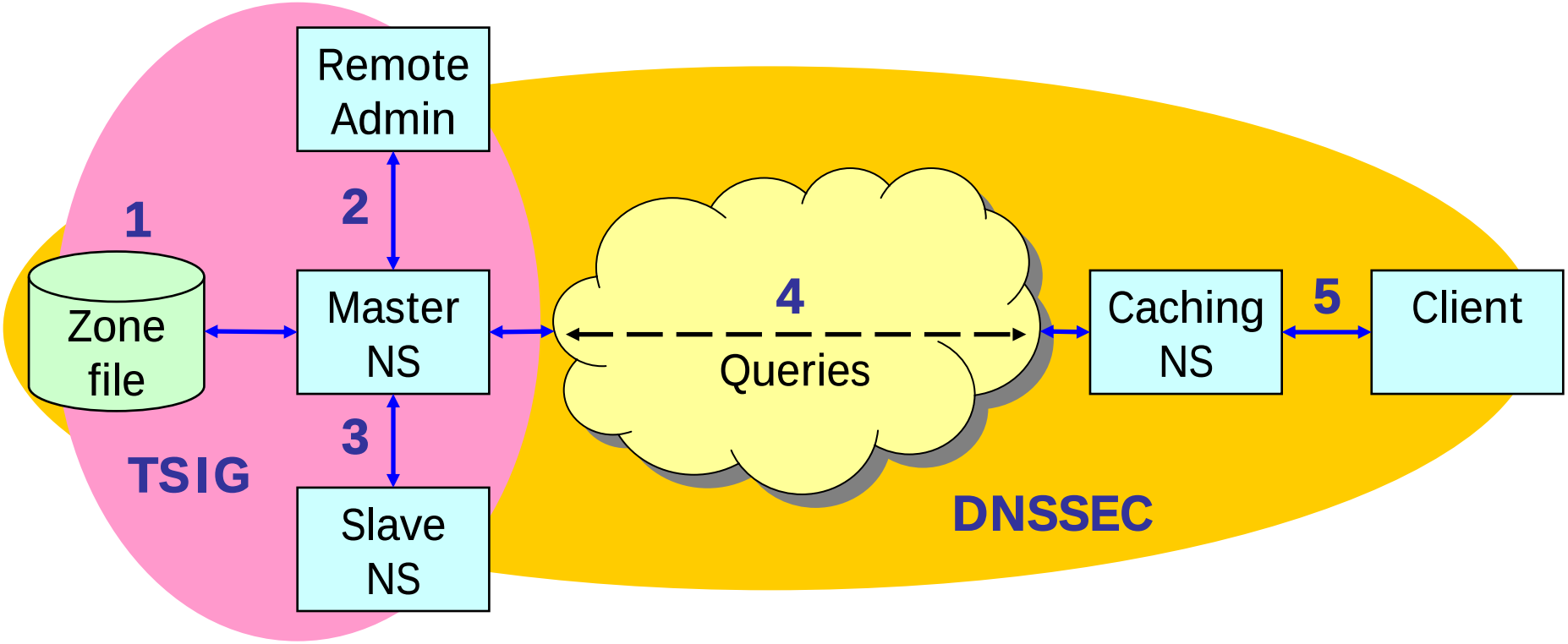
Johannes Kepler University Linz, Austria

sonntag@fim.uni-linz.ac.at

DNS security considerations

- DNS is a very important service and therefore a prime target
  - Imagine e.g. redirecting requests to "amazon.com" to your own web server....
  - Denial of service: Almost every service uses DN and not IPs!
  - Gathering network information: Which computers exist, guess their function from their name, impersonate them, ...
- Even more important when considering replication:
  - Many (esp. secondary) NS receive their data from other NSs
    - What about modifying/preventing this transfer?
- Security measures:
  - The server itself: Normal computer security
  - The clients: Ensuring they receive correct answers (and only those)

DNS security considerations



1: Zone file: Corruption, modification (local administration)
2: Dynamic updates: Unauthorized, IP address spoofing (TSIG)
3: Zone transfer: IP address spoofing (TSIG)
4: Remote queries: Cache poisoning, interception, subverted master/slave (DNSSEC), DDoS
5: Resolving: (As remote queries), IP spoofing (DNSSEC)

TSIG

- Shared secrets + hash function
- Advantages: Simple to configure, lightweight, flexible
- Main problems: Not scalable (shared secrets!), no key exchange (brute force attacks) → Requires regular manual key changes
- Usage:
  - Communication between name servers (zone transfers etc.)
  - Dynamic updates (DHCP → DNS), administration
- How it works: HMAC, not really signatures
  - Create hash value of data + shared secret and sent it along the data
    - Data: DNS response, time (→ replay attacks!), …
  - Recipient adds shared secret to data, hashes, and compares to received value

Signing DNS: DNSSEC

- DNSSEC= DNS SECurity
  - For secure communication between nameservers and clients
- Rarely used now, but currently transition to full support starting from root
  - Root has been changed (May-July 2010), now country-code TLDs are porting
    - DENIC: Will start 31.5.2011 (after extensive testing phase)
      - Note: This doesn't mean that everything under .de is signed, only that bmw.de, audi.de… will be signed
        - www.bmd.de, ftp.audi.de → The companies or their ISP must introduce DNSSEC as well!
    - nic.at: Currently only a testbed is available (with fake data)
      - http://www.nic.at/de/service/technische_informationen/dnssec/

Signing DNS: DNSSEC

- Solves: Integrity and authentication
- Limitations:
  - No encryption: All transmitted data is public
  - No protection against DoS, buffer overruns, etc.
  - No identification of clients (→ But this is not needed/desirable anyway!)
- Remaining problems:
  - Every client must perform the full validation itself (rarely done; normal PCs can't do it)
    - Else you have to trust your caching NS completely, that it does the verification!
  - Key rollover is complicated; no procedure at all for root key
  - How to securely (and automatically!) update key in parent zone when it changes?
  - Responses don't fit into single packets any more → TCP resolving needed

DNSSEC Requirements

- Must be able to securely confirm the existence of a record, and who it came from
  - "www.example.com" → "192.168.1.1"
- Must be able to securely confirm the absence of a record
  - "www.xeample.com" → "Does not exist"
- Must be able to securely provide answers only to the questions asked
  - "www.example.com" → "192.168.1.1 and incidentally www.bank.com" → ~~"~~10.0.0.1"
- Must allow distributed management (= no single authority producing all keys)
  - Every name server must be able to administer its own data itself
- Should not require too much ressources (server load)
- Little modifications, backwards compatibility

Signing DNS: DNSSEC

- DNSSEC is a public-private key method
  - The master nameserver signs the data with his private key
  - Any client can verify the data authenticity
- How do you find the master nameserver? Look one level higher!
- Termination of chain: DNS root
  - This cannot be verified and you just have to know the public key!
- Note: Parents do not sign the child zones itself!
  - The parent signs a pointer to the key (KSK – Key Signing Key) used for signing the key used for signing the child zone (ZSK – Zone Signing Key)
    - Important for organization!
  - Therefore: No certificates to be found anywhere (different kind of hierarchy)!

New resource records

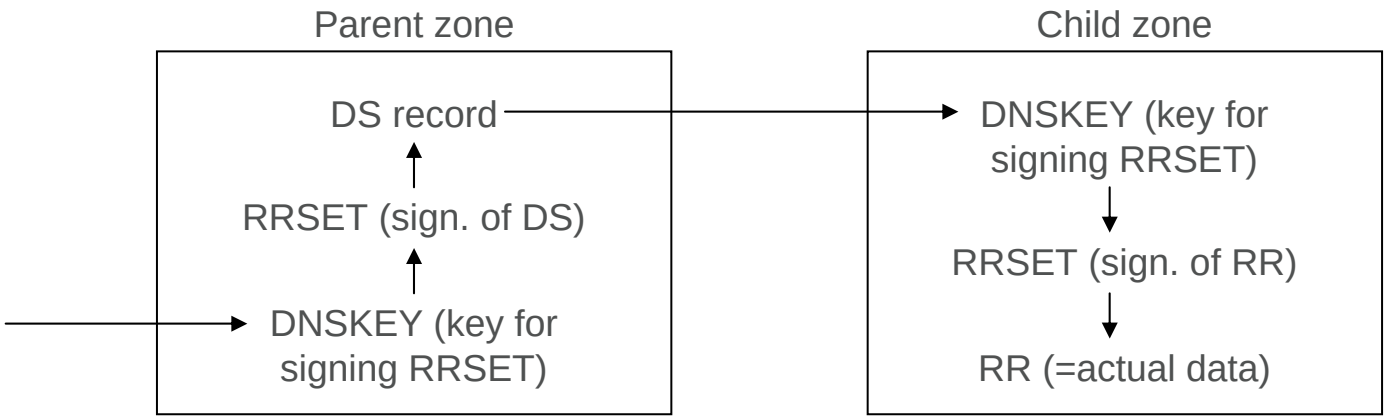
- DNSKEY: Public key used for zone signing (ZSK)
  - Our own key; RSA/SHA1 is mandatory, DSA/SHA1 optional
- RRSIG: Signature for a set of resource records
  - Contains signature expiration date (typ. 30 days)
  - Keys don't expire, only the record signatures → Must be updated regularly or the zone disappears from the Internet!
- DS: Delegation Signer. Two hashes (SHA1 and SHA256) of the (same) public key used to sign the key which is used for signing the zone data
  - To allow traversing the tree; i.e. the hash of the KSK of the next level below
  - Indirection introduced to allow a zone to change its own key regularly without requiring the parent zone to sign this key every time anew

New resource records

- NSEC: Used for secure confirmation of absence
  - Record are precalculated and not signed “on request” → “Strange” queries cannot be anticipated and therefore not pre-calculated!
  - NSEC tells which is the next record
    - Records are sorted lexicographically (myzone. → sub.myzone. → test.myzone.)
    - Two record + NSEC → What would be in between doesn’t exist!
  - Last NSEC record points back to first → Closed circle
  - Even more complex because of wildcard records (“*.myzone.com. → 10.0.0.1”)
  - Problem: Allows complete enumeration of a domain (→ Privacy issue)
    - Reduced by dynamically signing responses (has its own problems!) or NSEC3 (salted + hashed version of all possible names to deny, that other names exist)

Trust relationship

- Simplified version: ZSK only, no KSK!
  - Data (RRs) is trusted, if it is signed with a valid key (signature is in RRSET record)
  - The key is stored in the DNSKEY value
  - The DNSKEY is trusted if a matching DS record points to it from the parent
  - The DS record is trusted if it is signed by a trusted key (within the parent)



Political aspects

- Extreme power over the Internet in the had of the person owning the root key
  - Currently, this is the ICANN
  - Creating the key is done only by Verisign
  - Both root keys are stored in the USA only (west- and east-coast)
  - Department of Homeland Security requested that keys are owned by US government
  - “Trusted Community Representatives” are involved for accessing the hardware security module for creating the key as well as for recovery (each has only part of the credentials required → several are needed)
    - CO (Crypto officer; key generation, 7 per secure facility): Physical key to a safe deposit box located within the secure facility

Conclusions

- Will still take a longer time till the whole DNS (or at least the most important ones) are fully (complete chain to root) signed
  - Complex to administer → Bad for individuals/small corporations
  - Will probably done only by ISPs/large corporations anymore
- Often requires client modifications as well (web browsers do their own lookup → They must do their own validation!)
  - Also, requires trusting the resolver who actually verifies; this is typically not the client
    - The communication with this resolver must be secured as well (TSIG, …)
- A good step, but not that easy to finish!
- DNS can be used for all kinds of new records now
- Signatures in a completely different way: Without a PKI (but again hierarchically!)

Thank you for your attention!

Michael Sonntag

Institute for Information processing and microprocessor technology (FIM)

Johannes Kepler University Linz, Austria

sonntag@fim.uni-linz.ac.at

Literature

- Phil Regnauld, Hervey Allen, DNSSEC Deployment – A Tutorial (2009)
  http://nsrc.org/tutorials/2009/apricot/dnssec/dnssec-tutorial.pdf
- RFC 4033: DNS Security Introduction and Requirements
  http://www.rfc-archive.org/getrfc.php?rfc=4033