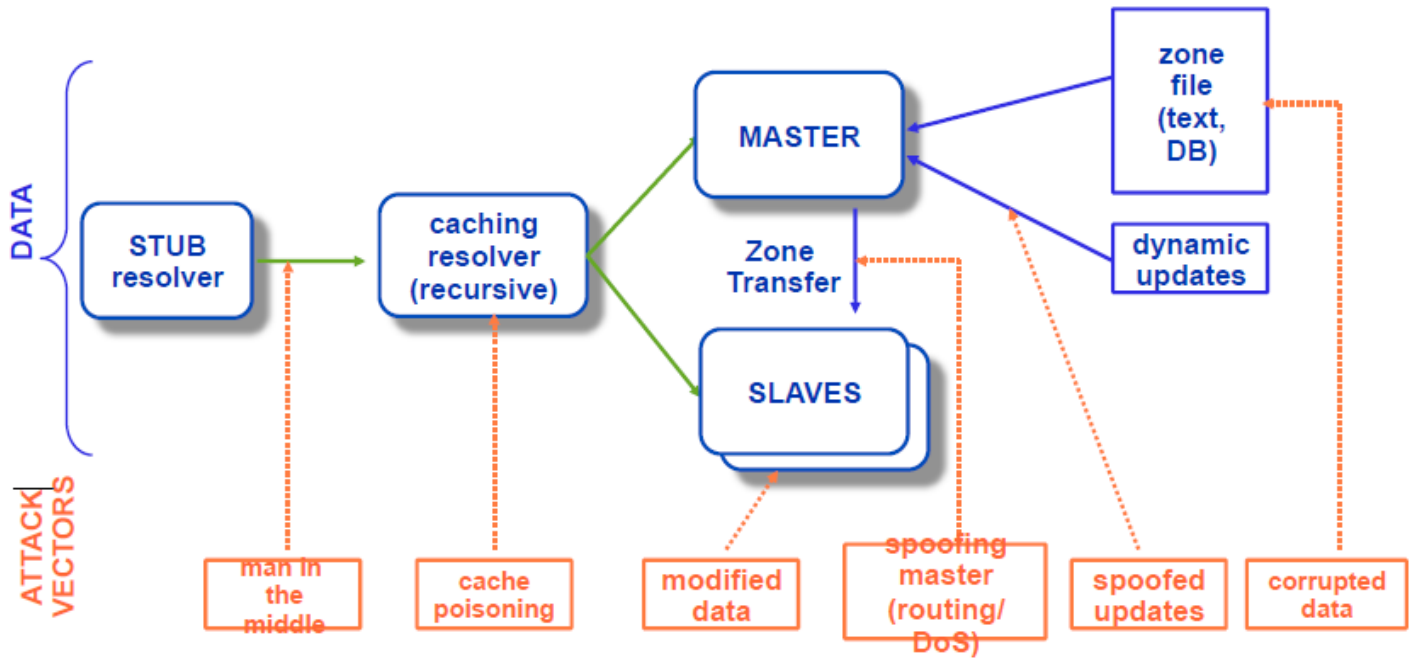
DNSSEC

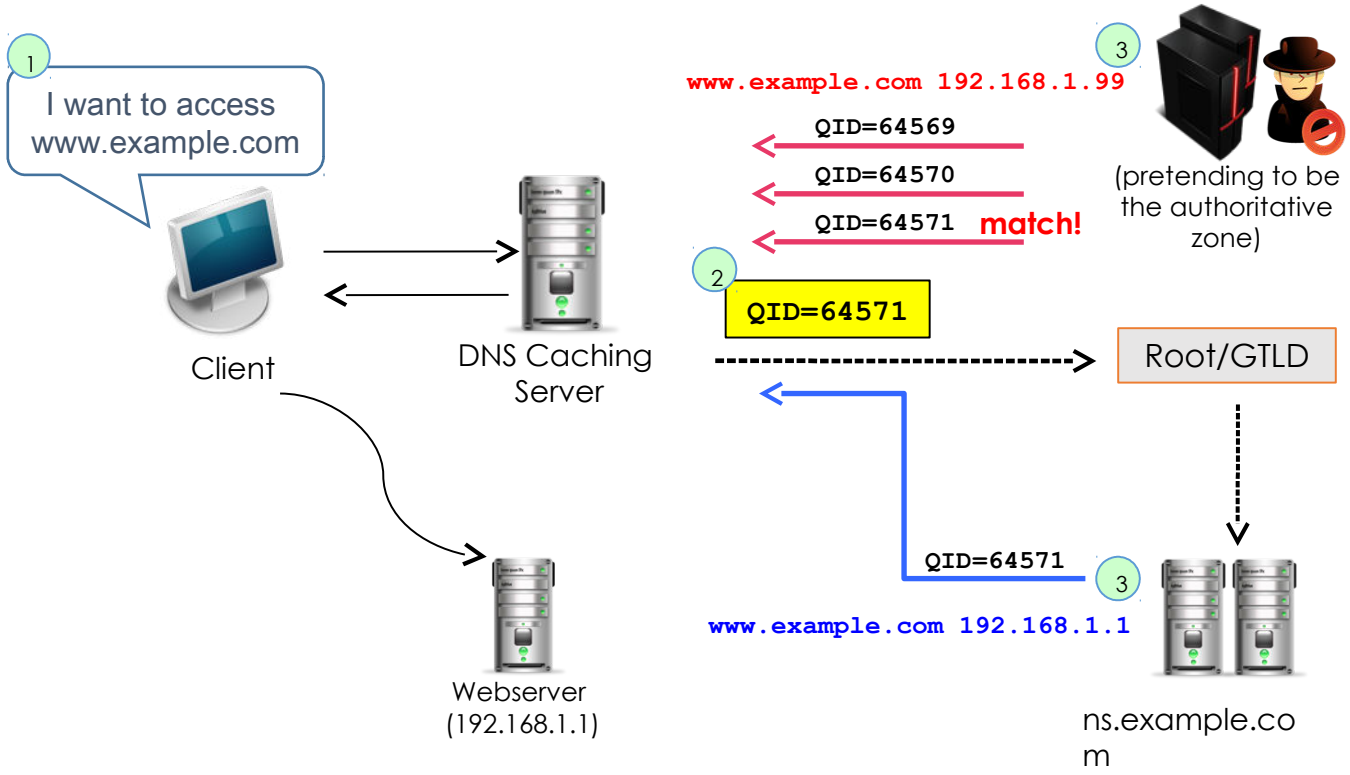
Keiichi Shima <keiichi@iijlab.net>
Slides stolen from Alisha Gurung

DNS Data Flow

Points of attack



DNS Cache Poisoning



RISKS

- **Misdirection of queries for an entire domain(Cache Poisoning)**

– Response to non-existent domains

– MX hijacking

– Make a large domain (SLD or TLD) domain

“disappear” from an ISP's cache – DoS

– Identity theft using SSL stripping attacks

(banks, eGovernance)

- **Many more**

DNS DOS Attack

- **DNS amplification**
- **→A type of reflection attack combined with amplification Source of attack reflected off another machine**

- **Traffic received is bigger (amplified) than the traffic sent by the attacker UDP packet's source address is spoofed**

Eg: (Spamhaus Attack)

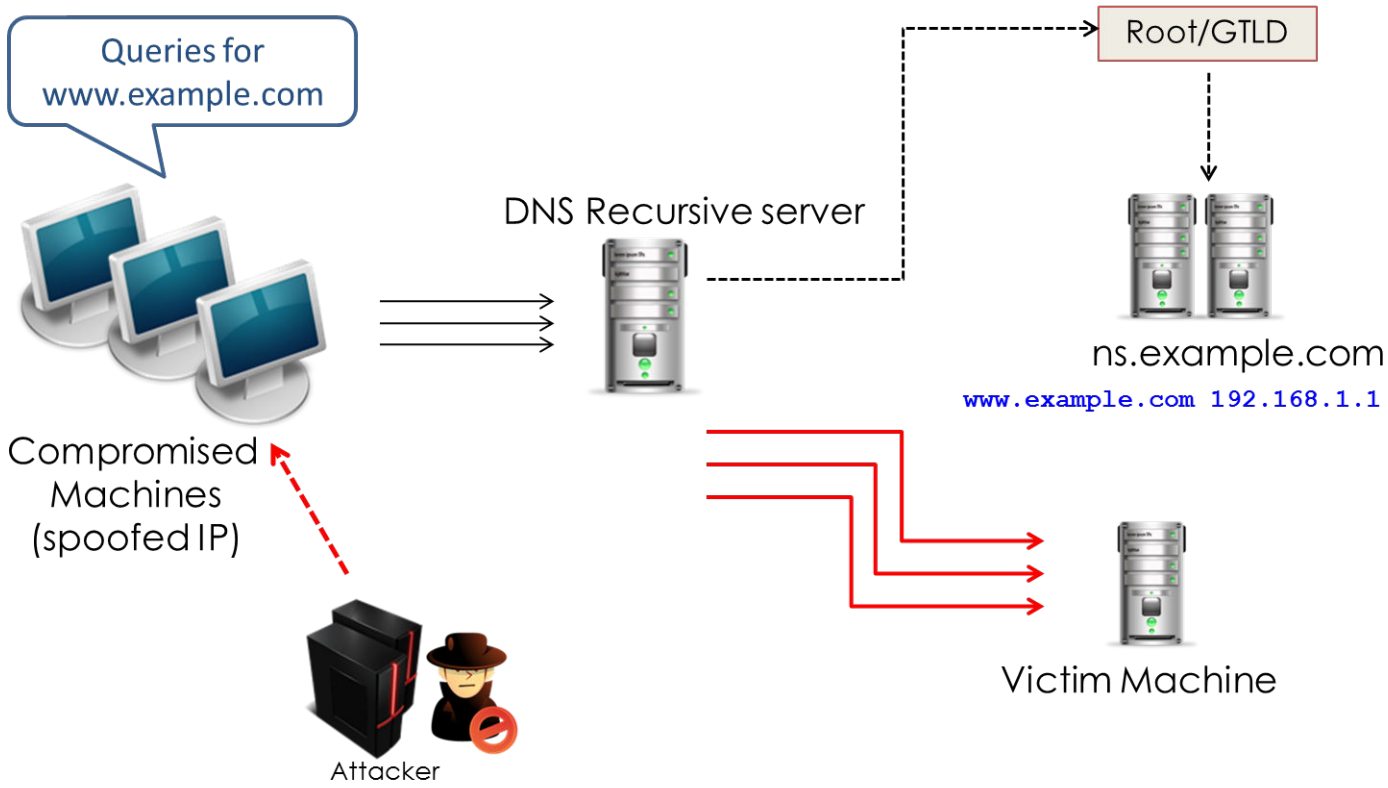
- **DNS query floods**

→"flood of legitimate-seeming queries sent to target DNS server for a domain"

→Uses botnet that automatically sends a significant number of queries

→Difficult to differentiate between a standard and malicious query

DNS Amplification



Open Resolvers

- DNS servers that answer recursive queries from any host on the Internet

  →http://openresolverproject.org/

- Check if you're running open resolvers

  →http://dns.measurement-factory.com/cgi-bin/openresolvercheck.pl

- More statistics at

  →http://dns.measurement-factory.com/surveys/openresolvers/ASN reports/latest.html

Good DNS Practices

- Enter the correct e-mail address of the responsible person for each zone you add to or manage on a DNS server.
- Do not combine authoritative and recursive nameserver functions -- have each function performed by separate server sets.
- Run multiple, distributed authoritative servers, avoiding single points of failure in critical resource paths. A variety of strategies are available (including anycast and load-balancing) to ensure robust geographic and network diversity in your deployment.

Securing NameServers

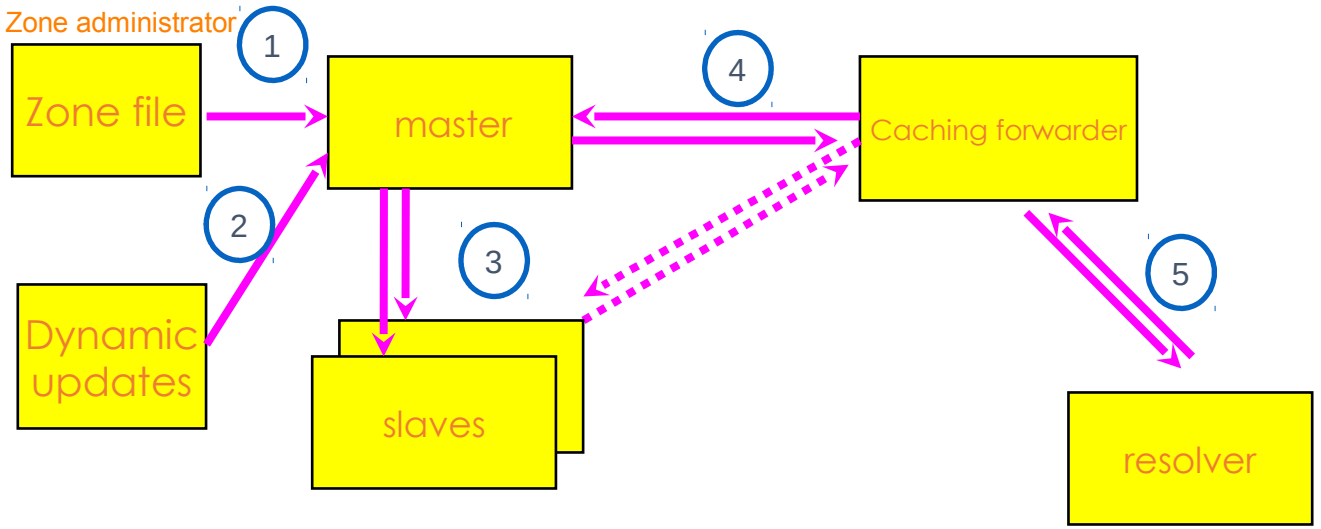
- Run the most recent version of the DNS software

  →Apply the latest patch
- Hide version
- Restrict queries

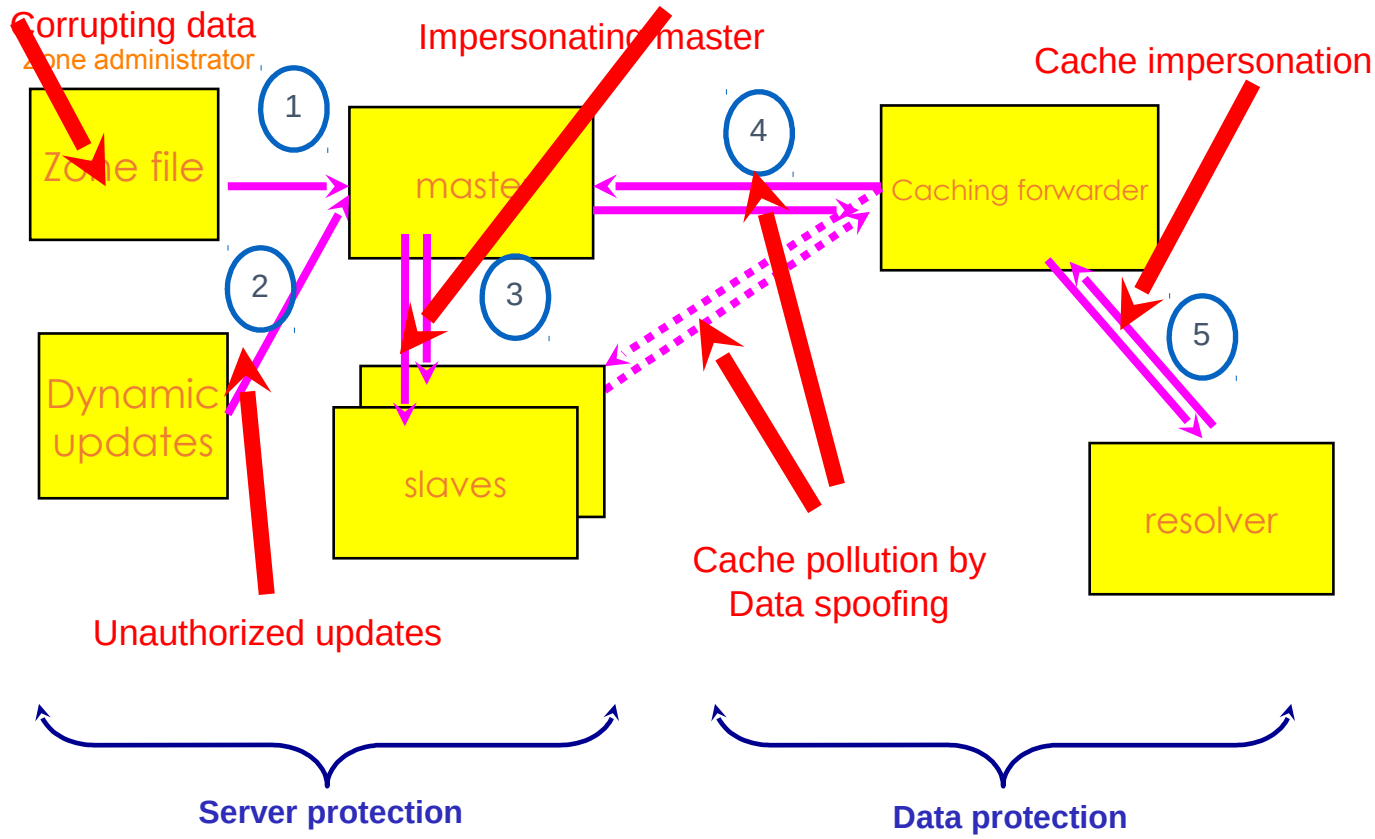
  →Allow-query { acl_match_list; };
- Prevent unauthorized zone transfers

  →Allow-transfer { acl_match_list; };
- Run BIND with the least privilege (use chroot)
- Use TSIG and DNSSEC

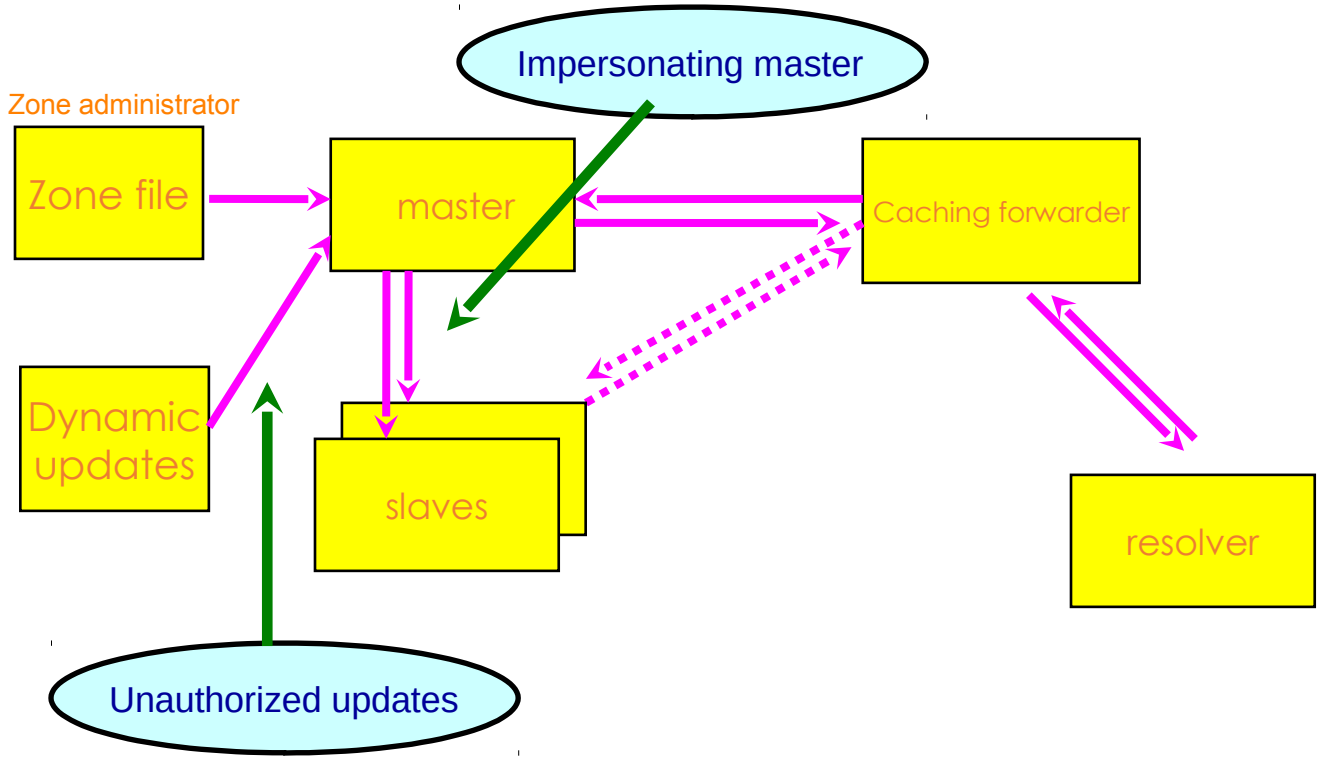
DNS: Data Flow



DNS Vulnerabilities



TSIG Protected Vulnerabilities



What is TSIG - Transaction Signature?

- A mechanism for protecting a message from a primary to secondary and vice versa
- A keyed-hash is applied (like a digital signature) so recipient can verify message
  - →**DNS question or answer**
  - →**and the timestamp**
- Based on a shared secret - both sender and receiver are configured with it

Transaction Signatures (TSIG)

- TSIG is most-commonly used to authenticate slave servers to master servers during zone transfers

  →Protects against impersonating master and unauthorized updates

- Master and slave servers:

  →share a common secret key & agree on key name

  →Synchronized clocks (NTP)

- The shared information (key) is used to authenticate a client to a server

  →Remember to change the key periodically

What is TSIG - Transaction Signature?

- TSIG (RFC 2845)
  - →authorizing dynamic updates & zone transfers
  - →authentication of caching forwarders

- Used in server configuration, not in zone file

TSIG steps

- **Generate secret**

```
dnssec-keygen -a <algorithm> -b <bits> -n
 host <name of the key>
```

- **Communicate secret**

```
scp <keyfile> <user>@<remote-server>:<path>
```

  - **Configure servers**

```
key { algorithm ...; secret ...;}
server x { key ...; }
```

- **Test**

```
dig @<server> <zone> AXFR -k <TSIG keyfile>
```

DNS Security Extensions (DNSSEC)

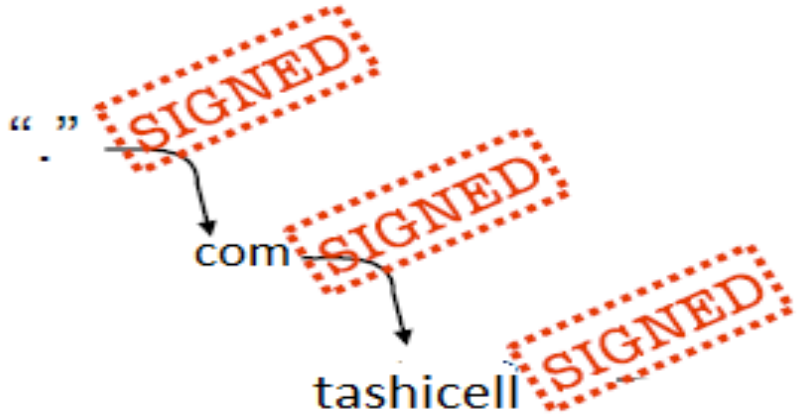
- Protects the integrity of data in the DNS by establishing a chain of trust
- Uses public key cryptography – each link in the chain has a public/private key pair
- A form of digitally signing the data to attest its validity
- Standard is defined in RFC4033, RFC4034, and RFC4035
- Guarantees
  - **Authenticity**
  - **Integrity**
  - **Non-existence of a domain**

DNSSEC Concepts

- Changes DNS trust model from one of ”open” and ”trusting” to one of ”verifiable”
- Use of public key cryptography to provide:
  - → Authentication of origin
  - → Data integrity
  - → Authenticated denial of existence
- No attempt to provide confidentiality (NO encryption)
- DNSSEC does not normally place computational load on the authoritative servers (!= those signing the zone)
- No modifications to the core protocol.

DNSSEC Concepts

- Build a chain of trust using the existing delegation-based model of distribution that is the DNS



DNSSEC Concepts

- Don't sign the entire zone, sign a RRset

  →Note: the parent DOES NOT sign the child zone.

- The parent signs a pointer (hash) to the key used to sign the data of child zone (DS record).
- Always on the master server.
- Check if slaves are receiving the signed zones.

DNSSEc: new 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
- **NSEC** = Returned as verifiable evidence that the name and/or RR type does not exist

- DS record provides a mechanism to delegate trust to public keys of third parties

Types of Keys

- **Zone Signing Key (ZSK)**
  - →Sign the RRsets within the zone
  - →Public key of ZSK is defined by a DNSKEY RR

- **Key Signing Key (KSK)**
  - →Signed the keys which includes ZSK and KSK and may also be used outside the zone

Signing the zone
(using the BIND tools)

1. Generate keypairs
2. Include public DNSKEYs in zone file
3. Sign the zone using the secret key ZSK
4. Publishing the zone
5. Push DS record up to your parent(how?)

(Note:signing always done in Master and validation in resolver)

Key Generation

Generate ZSK

dnssec-keygen [-a rsasha1 -b 1024] -n ZONE myzone

Generate KSK

dnssec-keygen [-a rsasha1 -b 2048] -n ZONE -f KSK

myzone

- **Signing Of Zone**:dnssec-signzone myzone

KEY MANAGEMENT

- Need to implement secure key storage, management procedures
- Need to sign your zones
- Registries need to accept DS records from users (how?)
- Need to publish DS records to parents (how?)
- Manual Signing: Key RollOvers
- Key Never expires. Need to resign
- Automated Signing: Opendnssec or inline signing.

Inline Signing

- **Dnssec Signing made easy.(automatic signing and key rollovers)**
- **Requires Bind Version 9.9 and above.**
- **Enabled by the "inline-signing yes;" statement in named.conf**
- **Create zsk and ksk but sign using inline and resign using (auto-dnssec maintain).**
- **Check if it's signing.**
- **Check logs.**
- **$ dig @localhost mytld NS +dnssec(to check if dnssec is working, for manual also)**
- **$ sudo named-checkzone -D -f raw -o - mytld mytld.signed | less**
- **how do we update the zone and resign it ?**
- **Problem:Doesn't generate keys in some distro,use ""haveged".**

Inline Signing/ Bump in the Wire