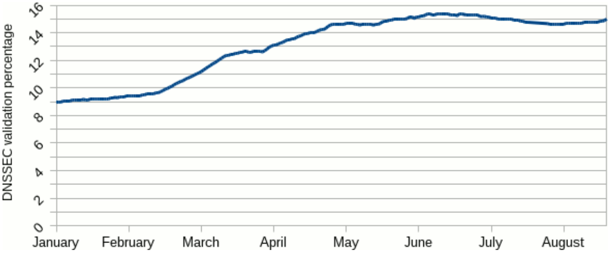
Challenges in endpoint DNSSEC

Ondřej Caletka

November 2014

Current implementation of DNSSEC



Source: `http://stats.labs.apnic.net/dnssec`

- DNSSEC-based apps are being developed
  - SSHFP (SSH server public key fingerprint)
  - DKIM (e-mail authentication)
  - TLSA (DANE TLS certificate pining)
- almost nobody does the validation correctly

  *Trusting the AD-flag from nearby DNS server over untrusted network is **wrong**.*

DNSSEC validating home router Turris

- shorten the insecure first mile to the users’ homes
- deployed in ~1000 households across Czechia
- uses Unbound in forwarding or full recursion mode
  - forwarding mode **almost never works well**
  - for few ISPs, **even full recursion does not work**

Forwarding mode problems

- known bug of BIND versions < `9.9` in recursive mode
- all DNS names synthetised from wildcards are seen as bogus
- users tend to *“blame the postman”*
- fixed in current upstream stable versions
- it will take years until ISPs get rid of old broken versions of BIND

Full recursion mode problems

- **it does not scale well**
- DNS traffic engineering, especially with small ISPs
  - DNAT everything udp/53 to ISP’s DNS server
  - DNAT everything udp/53 to 8.8.8.8
  - *“Nobody’s complaining, so what’s the problem?”*
- Various “security features” like DNS inspection:
  - droping udp/53 packets bigger than 512B
  - Cisco hint:
    `inpect dns maximum-length 4096`

DNS64 vs. DNSSEC

- new challenge for endpoint validation
- synthetic AAAA records from DNS64 cannot be DNSSEC validated

⇒ you have to trust the `AD` flag from DNS64 device

- or do DNS64 at your localhost after DNSSEC validation
- problematic full recursion mode due to IPv4-only nameservers (*even Google*)

Provisioning localhost DNS64

- RFC 7051 proposes a few solutions:
  1. DNS Query for a Well-Known Name
  2. EDNS0 flags or options
  3. DHCPv6 option
  4. RA option
  5. Application layer protocol like STUN
- RFC 7050 describes solution no. 1:
  1. query for WKN `ip4only.arpa IN AAAA`
  2. use heuristics to find out NAT64 prefix
- automatic discovery opens some new attack vectors (redirecting all traffic to certain IPv6 prefix), *if not done properly*

Conclusion

- deploy DNSSEC validation on your DNS recursors
  if Google can do it, you can as well
- don’t block or redirect udp/53 packets of any size
- when deploying NAT64, prefer using well-known prefix, *they are harder to misuse*
- when using network-specific prefix for NAT64, make sure you set up DNS in a way it allows prefix validation (see RFC 7050)

Any Questions?

Ondřej Caletka
`Ondrej.Caletka@cesnet.cz`