AMRES
Serbian Academic Network

BELGRADE SECURITY WORKSHOP 2015

DNSSEC operational experiences and recommendations

Antti Ristimäki, CSC/Funet

GÉANT

Agenda

- Funet DNSSEC status
- A short DNSSEC tutorial
- Zone signing considerations
- Private key security
- Network layer impacts
- Monitoring

Funet DNSSEC status

- DNS resolvers doing DNSSEC validation since January 2012
- Own and hosted customer zones not signed yet
  - signing system development on hold due to a lack of human resources and relatively low customer demand
- Finnish ccTLD (fi.) master server A.fi operating outsourced to CSC/Funet since 2007
  - since 2010, also including the DNSSEC signing system
  - operational experience from multiple key rollovers and one complete signing system migration last autumn
- BCP document “Guidelines for deploying DNSSEC” written for GN3 campus project

GN3+ Campus Networking BCP

Guidelines for Deploying DNSSEC

Best Practice Document

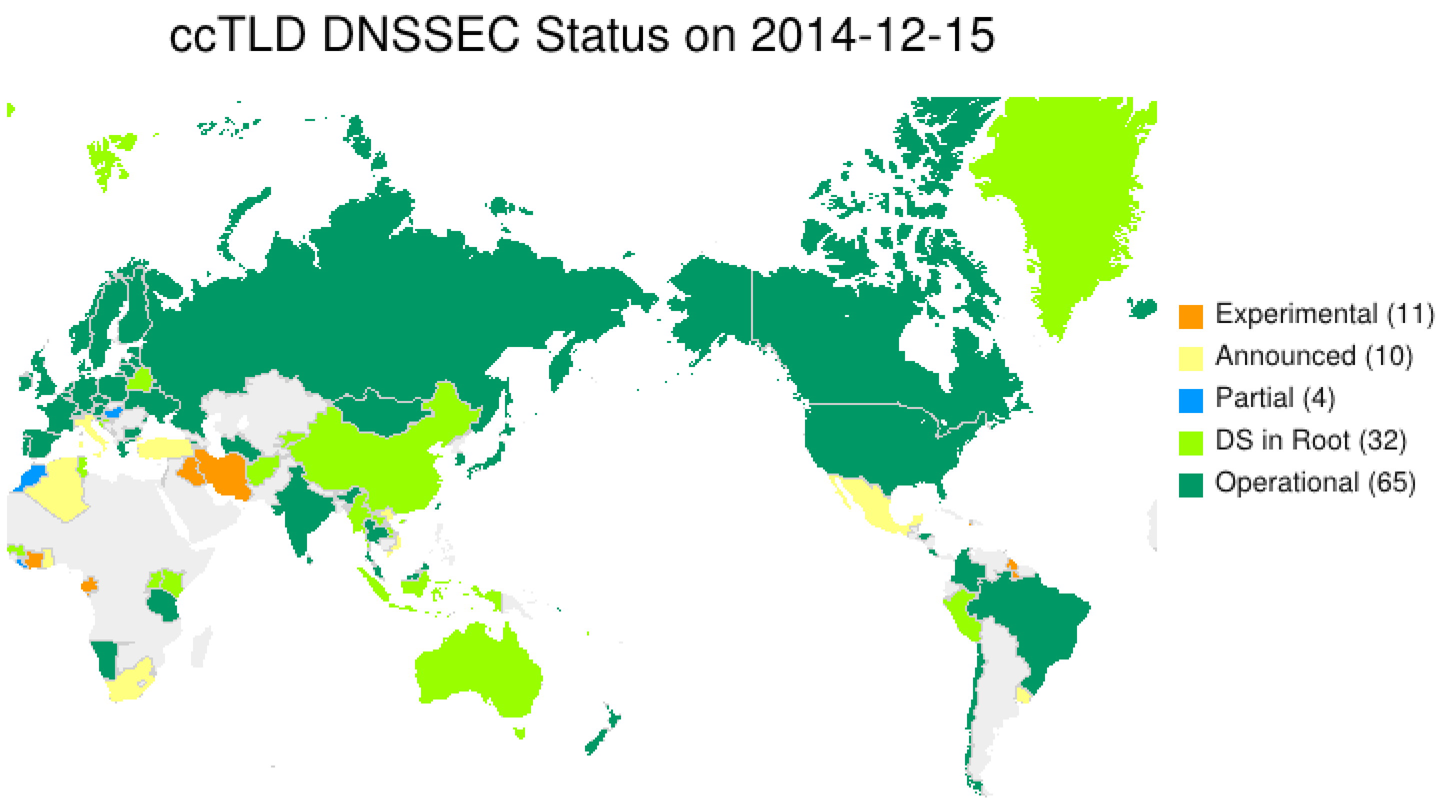
Produced by the CSC/Funet-led working group on AccessFunet

Authors: Antti Ristimäki (CSC/Funet), Tuukka Vainio (University of Turku), Kaisa Haapala (CSC/Funet)

DNSSEC in a nutshell

- DNS Security Extensions
  - core RFC documents 4033-4035
- Developed primarily to prevent DNS spoofing
  - deployment accelerated after Kaminsky bug (2008)
  - internet root zone signed in July 2010
- DNS resource records digitally signed
  - data origin authentication
  - data integrity
  - authenticated non-existence
- Based on public key cryptography
  - DNS records signed with a private key and validated by a corresponding public key
- Public Key Infrastructure built within the DNS itself
  - public keys and their signatures distributed in DNS
  - chains of trust follow the DNS hierarchy

DNSSEC deployment status



source: www.internetsociety.org

DNSSEC Public Key Infrastructure

- DNSSEC public keys published in DNS in form of DNSKEY record, no need for key servers etc.
  - scalability
- A digest (DS record) of the public key published in the corresponding parent zone and signed with parent zone's private key
- This chain continues up to the top of the DNS hierarchy, i.e. the signed root zone
  - root zone's public key usually obtained by some out-of-band method and pre-configured as a "trust anchor"
- In practice, usually two separate key pairs used within a DNS zone
  - Zone-Signing Key for signing the zone DNS data
  - Key-Signing Key for signing only the key set – must be synchronized with the information in the parent zone

An example

zone nxdomain.fi:

server.nxdomain.fi.	402	IN	A	84.20.146.223
server.nxdomain.fi.	402	IN	RRSIG	A 8 3 600 20150411044315 20150312044315 **2043** nxdomain.fi. *Act0uYd+jK*…..

nxdomain.fi.	266	IN	DNSKEY	256 3 8 *AwEAAcw*……; ZSK; alg = RSASHA256; **key id = 2043**
nxdomain.fi.	266	IN	DNSKEY	257 3 8 *AwEAAbV*……; KSK; alg = RSASHA256; **key id = 2332**
nxdomain.fi.	266	IN	RRSIG	DNSKEY 8 2 600 20150411044315 20150312044315 **2332** nxdomain.fi. *IU5IqRLO1SPl*…..

zone fi:

nxdomain.fi.	18773	IN	DS	**2332** 8 2 *BFED9CF2C1E2C386CDC5E368282431E3871FBC65ABEC8D714B827722 1AC4117D*
nxdomain.fi.	18773	IN	RRSIG	DS 8 2 21600 20150327235351 20150313171012 **1404** fi. *ZbiLGXPdLyX*……

fi.	402	IN	DNSKEY	256 3 8 *AwEAAeOS*…….; ZSK; alg = RSASHA256; **key id = 1404**
fi.	402	IN	DNSKEY	257 3 8 *AwEAAboRq*…..; KSK; alg = RSASHA256; **key id = 28547**
fi.	402	IN	RRSIG	DNSKEY 8 1 900 20150326093719 20150312001012 **28547** fi. *fCyLQBdR4f*….

zone . (root)

fi.	83270	IN	DS	**28547** 8 2 *F93C02BA667174063450993218844E1AFBB402E24118FAC03D9F2234A 6A95B846*
fi.	83270	IN	RRSIG	DS 8 1 86400 20150325170000 20150315160000 **16665** . *YAAlcr6zc4m*……..

.	155109	IN	DNSKEY	256 3 8 *AwEAAe3*……; ZSK; alg = RSASHA256; key id = **16665**
.	**155109**	**IN**	**DNSKEY**	**257 3 8 *AwEAAag*……; KSK; alg = RSASHA256; key id = 19036**
.	155109	IN	RRSIG	DNSKEY 8 0 172800 20150326235959 20150312000000 19036 . *K120+hpWw*…….

DNSSEC validation (signature verification)

- Validation done by resolvers (if configured to do so)
    - recursive and forwarding name servers, stub resolvers
- Validating resolver tells the authoritative name server that it is willing to receive DNSSEC related data
    - EDNS0 extension with “DNSSEC OK” bit in pseudo-header
- Validating resolver marks validated answer with “Authenticated Data” header bit
- Ideally, signature validation would be done as close to the end user as possible
    - currently poor DNSSEC support in OS resolver libraries
    - some enthusiastics run a local validating DNS resolver (e.g. Unbound) in localhost
    - for example Dnsmasq nowadays supports DNSSEC validation
- Today signature validation usually done by ISP’s resolvers
    - connection between the client and the resolver still possible spoofing point

DNSSEC validation (cont.)

```
$ dig +nocmd www.dnssec.fi a +dnssec +multiline
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24941
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.dnssec.fi.                IN A

;; ANSWER SECTION:
www.dnssec.fi.                 214 IN A 87.239.122.38
www.dnssec.fi.                 214 IN RRSIG A 8 3 300 (
                                    20150615085607 20150211085610 1947 dnssec.fi.
                                    cghIwaCTENXg8WcwZTEqfdLJ1CJLfVVWNlmUuq7WCLkC
                                    j0Mt4L9jDIzke0kmOf/B30D2s3pnuqz6JVeFb2ZReZsG
                                    LW9Q9Qccve9hF0oSRoHFnWh6QKwxzKdZWgOf+FcAMH7f
                                    74IYmP9endI56hs+Zp6Hd2itCHSKE3uSg5AfW7g= )
```

AD = authenticated data

DO = DNSSEC OK

Zone signing

- “Bump in the wire” type of solutions exist, both open-source (e.g. OpenDNSSEC) and commercial
  - Zone-transfer (AXFR), SSH/SCP or some other mechanism for receiving the unsigned zone and for distributing the signed zone to authoritative DNS servers
  - deployment requires a lot of effort, but can make up to a very configurable, customizable and secure solution
- Some DNS management software also support DNSSEC signing
  - easy deployment, usually only a tick in a checkbox
  - varying levels of implementation quality/robustness
- Authoritative DNS server software with automatic inline signing support exists
  - at least Bind, KnotDNS (beta)
  - easy deployment, usually only a couple of lines of configuration
- Early deployments based on custom scripts built around “dnssec-signzone”
- Building a redundant signing system requires thorough planning
  - the same set of keys must be available for all servers
  - state synchronization also needed: which keys to use, when to roll over, which SOA serial to write into the signed zone…

Private key security

- Keeping private keys private is the basis for the whole security of DNSSEC
- Desired level of protection is usually determined by the importance of the DNS zone to be signed (e.g. Top-Level vs Second-Level domain)
  - Hardware Security Modules, usually double as “crypto accelerators”
  - USB tokens
  - encrypted or in plain-text on disks
- Keys should be truly random
  - good random number generator needed!
  - CPU on-chip hardware RNG (e.g. Intel RdRand) might be worth looking at
- Don’t forget to backup your private keys
- Strong algorithms should be preferred, with resolver support taken into consideration
  - RSA/SHA-1, RSA/SHA-256 and RSA/SHA-512 widely used today
  - use of ECDSA also specified, resolver support questionable

Private key security (cont.)

- It is an usual practice to roll over keys at regular intervals in order to make brute force calculation more difficult
- Some choose not to perform any regular rollovers, but rely on strong algorithm and large key size
- IMHO, in a long run some kind of key rollover mechanism is needed anyway
  - computing power increases all the time and algorithm weaknesses are discovered every now and then
  - analogy e.g. with SSL/TLS, where SHA-1 hash algorithm is being gradually deprecated
- Dedicated signing software (e.g. OpenDNSSEC) can usually handle key rollovers and related timings in a smooth fashion
  - Zone-Signing Key rollover usually fully automatic
  - Key-Signing Key usually requires human intervention, when updating DS record in parent

Zone signing parameters

- Many configurable parameters affecting DNSSEC operation
  - key algorithms and key lengths
  - periodic signing interval
  - signature validity period and refresh interval
  - key rollover intervals, if any
  - TTL values
- Timing parameters directly affect how much time there is time to fix issues in different failure scenarios
  - if signatures are refreshed only just before their expiration, in case of signing system failure there might be only little time to fix it
  - administrators are not working 24/7, holidays and weekends must be taken into consideration
- DNSSEC also has some impact on zone SOA record values
  - for example SOA expire value should be set so that the zone expires altogether before signatures
  - it is better to let the zone expire on a single slave server than letting it serve expired signatures

Common issues with signed zones

- Expired signatures
- Signatures created with a wrong or non-existing key, reasons including:
  - failure in a key rollover process
  - in a redundant signing system backup system using different keys from the primary one

  → we have deliberately decided not to use any automatic failover, backup system activated only after manual checks
- Too liberal algorithm rollover breaking conservative resolvers
  - varying algorithm support and behaviour in different resolver software
- All in all, there are many ways to break a domain with DNSSEC
  - and because DNS uses caching, recovery time can be painfully long
  - most issues can be prevented with proactive counter-measures and comprehensive monitoring (more on these later)

Zone signing precautions

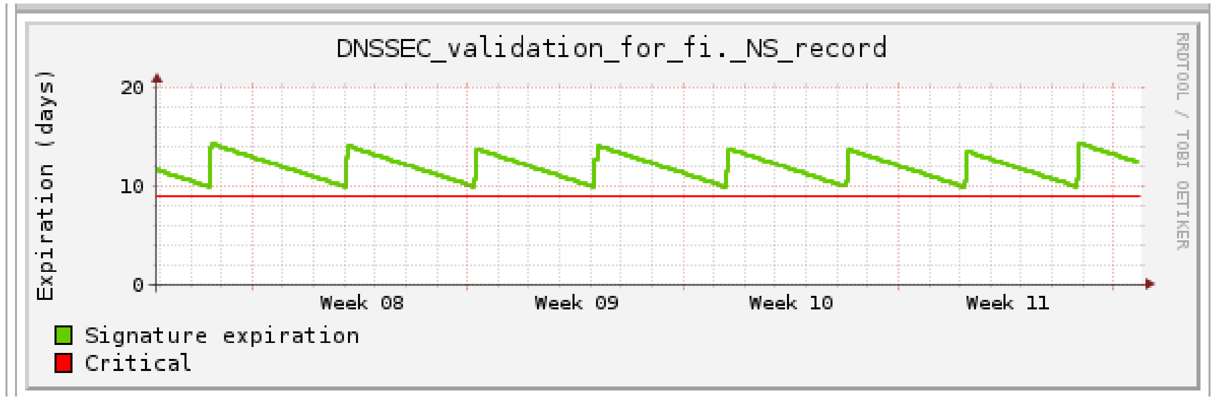
- Signed zone audit before publishment
- At least OpenDNSSEC supports running custom audit checks after signing
- In our environment, we feed the signed zone through Validns and also a couple of self-written validation scripts, ensuring that:
  - the chain of trust from the parent zone will remain intact
  - the signed zone can be validated with the DNSKEY existing in DNS caches
  - the information in DNS caches can be validated with the DNSKEY in the newly signed zone
  - the signing process hasn't dropped any data from the unsigned zone file
- These checks should make sure that an invalid zone file is **never** published

Network layer considerations

- With DNSSEC data, it is very common that an answer doesn't fit into a single UDP packet
  - IP fragmentation needed → make sure UDP fragments are not firewalled
  - running an authoritative DNS server with MTU higher than 1500 may call for problems
- Tunnels (VPN etc.), IPv6 and DNSSEC sometimes a difficult combination
- Especially with IPv6 it's important to not stop Path MTU Discovery from working by filtering ICMP error messages (Packet Too Big)
- It is also possible that the answer size exceeds the advertised UDP buffer size
  - resolver TCP fallback
  - many firewalls incorrectly configured to accept DNS queries/responses only over UDP

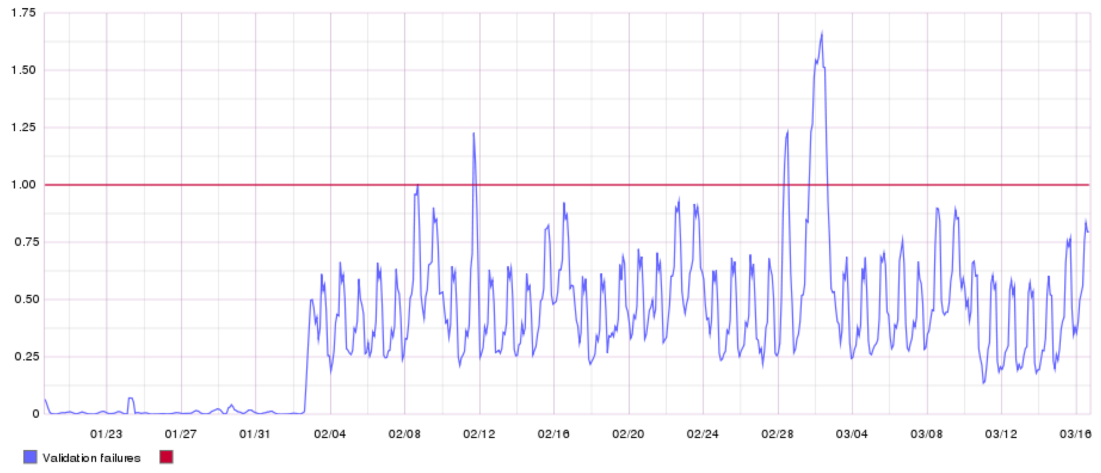
Monitoring

- DNSSEC makes proactive DNS monitoring far more important than before
- The most important thing is to be able to notice as soon as possible when issues start to evolve
  - reactive monitoring usually reveals issues when it's already too late
- For authoritative zones, it should be monitored that
  - the zone has been refreshed lately – this can be checked for example by inspecting the SOA record signature inception timestamp
  - the zone signatures are not going to expire any time soon – can be checked by inspecting the signature expiration timestamps
  - under normal circumstances, the signature remaining lifetime should constitute a saw-tooth graph



Monitoring (cont.)

- Time does matter in DNS, NTP sync monitoring very important
  - clock skew might affect validation results
  - important especially with virtual servers
- For validating resolvers, one should be able to notice when the rate of validation failures drastically increases
  - Funet resolver statistics collected in Graphite and the moving average value checked with Nagios via Graphite URL API
  - threshold has been increased a couple of times in order to avoid false positives



DNSSEC as a DDoS amplifier

- In presence of DNSSEC, a small DNS query can result in a very big answer
  - large amplification factor
- Authoritative DNS servers have been used a lot for DDoS amplification for example by querying “ANY” data with spoofed source address
- Most DNS server software nowadays support Response Rate Limiting (RRL) feature for mitigation

Conclusions

- DNSSEC can be hard and operating it successfully requires good knowledge, thorough planning and extensive monitoring
- Keeping things simple ensures smooth operation and helps debugging
- When properly deployed, DNSSEC could make DNS a feasible channel to publish for example
  - SSH host fingerprints (SSHFP record)
  - SSL/TLS certificates
  - PGP keys
- Lessons have been learned, operational practices and software hopefully gaining maturity

BELGRADE SECURITY

WORKSHOP

2015