DNSSEC. Chain of Trust

As described in our introductory article, DNSSEC is a technique that allows a DNS resolver to validate the contents of a DNS zone by using digital signatures. This makes tampering with the contents of a DNS zone (for example IP spoofing) impossible.

This article zooms in on the validation process that DNS resolvers perform. You will find information about:

1. Signing and validation of DNS records (DNSKEY and RRSIG records)
2. Involving the registry through the “chain of trust” (DS records)
3. Security at the highest level: the root zone

This article is built around an example: the DNSSEC-secured domain openprovider.nl. As DNSSEC keys and signatures are ever-changing, take the example as an “example” and don’t focus on the exact contents of each record. That’s highly likely to have changed since the moment of writing!

Signing and validation of DNS records (DNSKEY and RRSIG records)

Using a dig command, we can get the DNSSEC contents of the openprovider.nl DNS zone:

```
$ dig +dnssec @ns1.openprovider.nl openprovider.nl any

;; ANSWER SECTION:
openprovider.nl. 900 IN A 185.87.187.6
openprovider.nl. 900 IN RRSIG A
 8 2 900 20181004000000 20180913000000 47025 openprovider.nl. elN[...]948=

openprovider.nl. 1800 IN DNSKEY 256 3 8 AwE[...]WRd
openprovider.nl. 1800 IN DNSKEY 256 3 8 AwE[...]y3T
openprovider.nl. 1800 IN DNSKEY 257 3 8 AwE[...]A8=
openprovider.nl. 1800 IN RRSIG DNSKEY
 8 2 1800 20181004000000 20180913000000 5299 openprovider.nl. IYM[...]feg==
```

The starting point for validation are the **DNSKEY** records: those are the public parts of the cryptographic keys used to sign the zone. They are very important, as we use them to perform the validation!

Let's take a look at the **A record** in the first line, pointing to IP address **185.87.187.6**. A DNS resolver should use the public key from the DNSKEY record to check if the signature (stored in the **RRSIG record** at line 2) matches. If so, the A record is valid and 185.87.187.6 is the verified IP address.

Assume that someone tries to redirect you to a different IP address, e.g. 52.192.17.53. In that case, the signature will no longer match and the DNS resolver will return an error. You're protected against tampered records.

The same concept is used to check the validity of the DNSKEY records themselves: the last line in the above example contains the RRSIG signature for the DNSKEY records. The pitfall is that the signature of the key is checked with the key itself. This is an entry for a malicious change. So how can we be sure that the public keys are untampered with?

Involving the registry through the "chain of trust" (DS records)

The solution is to store a hash of the DNSKEY record at the registry, in a so-called DS record. If the hash matches the public keys in the DNS zone, we can be sure that those public keys are authentic and we can trust the signatures.

You can find the DS record by querying the registry's nameservers, in this case SIDN (the .nl registry):

```
$ dig +dnssec @ns1.dns.nl openprovider.nl

;; AUTHORITY SECTION:
openprovider.nl. 3600 IN DS 5299 8 2 A79[...]3AC
openprovider.nl. 3600 IN RRSIG DS 8 2 3600 20181005204824 20180921200803 38538 nl. QYx[...]zqg=
```

The shown DS record is a hashed version of the public key in the domain's zone. This principle is called the "chain of trust": we trust the domain's DNS zone because the parent (the registry) says it can be trusted.

But can we trust the contents of the registry's zone? Is the stored hash indeed valid, or has someone changed it? We can check this one level higher, at the root zone:

```
$ dig +dnssec @a.root-servers.net nl
```

```
;; AUTHORITY SECTION:
nl. 86400 IN DS 34112 8 2 3C5[...]D22
nl. 86400 IN RRSIG DS 8 1 86400 20181005050000 20180922040000 41656 . EIN[...]OGQ==
```

As long as this hash matches the key of the registry, everything is fine.

However, we must repeat our previous question: can we trust the contents in the root zone? Only if the answer is “yes”, we can trust the full chain.

Security at the highest level: the root zone

The problem is that the root zone has no parent. We cannot simply check a level higher - it does not exist!

That’s where we find the only human involvement: the security procedures around the keys of the root zone so very strict, that “we” (the internet community) have agreed that we can trust them, even though we cannot check them digitally at a parent.

By this agreement, we have completed the chain of trust: from domain name up to the root zone, every link is trusted and as such, the returned A record 185.87.187.6 can be trusted to serve the real and only openprovider.nl website.

Reading tip: curious towards the security measures around the root key management? Read here a first-hand report.

Revision #2
Created 2026-08-07 14:27:37 UTC by Sarath
Updated 2026-08-07 15:45:11 UTC by Sarath