DNS Resolver Documentation

Dhaval Bagal

September 2021

Contents

Chapter 1

DNS Resolver

1.1 DNS Implementation

The DNS resolver uses three main functions:

1. **resolve()**
2. **_get_resource_records()**
3. **_query_server()**

1.2 DNS Resolution

The project uses **resolve()** method to resolve domain names. The algorithm proceeds as follows:

Algorithm 1 resolve()

```
1: Filter hostname to remove all the non-zone elements like http, https, www, etc
2: Split the zones in the hostname.
3: n = number of zones
4: Initialize nameservers with a list of rootserver ip addresses
5: while iterating over all zones do the following do
       • Get rrsets (resource records) and IP of the queried nameserver using
         _get_resource_records() function while feeding the function a list of
         nameservers where the query needs to be redirected.
       • Update the next nameservers to be contacted by extracting the A/NS
         records from rrsets
6: In case of A and MX records, query the final authoritative nameserver to get the response.
7: end while
```

Algorithm 2 _get_resource_records()

```
1: Form a dns query object
2: Call _query_server() to forward this query object to the nameserver and get the response
3: Organize the response into a dictionary with the key as RR record type and value as the RRSet.
4: Return the dictionary along with IP of the actual nameserver queried from the list of the nameservers.
```

Algorithm 3 _query_server()

```
1: Query the first nameserver and if it fails go ahead with the second one and so on.
2: If nameserver contains a URL, resolve it to get its IP address
3: Return the query response along with the IP of the nameserver queried from the list of nameservers.
```

Chapter 2

DNSSEC Resolver

2.1 DNSSEC Implementation

This project implements the DNSSEC layer on the top of the DNS layer. So everything stays the same except for the fact that once the A record for a domain is fetched, **_check_trust()** is called to verify the chain of trust.

DNSSEC uses four main functions:

1. **_check_trust()**
2. **_verify_ksk()**
3. **_verify_zsk()**
4. **_verify_rrset()**

2.2 Verifying the chain of trust

Once the desired record for a domain name is fetched by the DNS, we need to check its validity.

For this, we first verify the ZSK that we got. We query the domain name with the DNSKEY record which returns the ZSK, KSK and RRSig. The RRSig is the ZSK signed using KSK. We verify the ZSK by signing it with KSK and comparing it with RRSig to see if its valid.

To check the validity of the KSK, we go up the chain and trace the route upwards and verify the KSKs for each zone. If all the KSKs up the chain are valid, then the KSK that we received in the DNSKEY record is valid too.

After validating KSK and ZSK, we validate the RRSet by signing it with ZSK and comparing it with the RRSig which in this case is the RRSet signed using ZSK.

Algorithm 4 _check_trust()

```
1: Query the FINAL authoritative nameserver for DNSKEY record
2: If there's no response =¿ DNSSEC is not enabled for the domain
3: Iterate through the redirection history
       • Query the parent for DS record
       • Extract KSK from the child DNSKEY response
       • Call _verify_ksk() to verify the extarcted KSKs using the DS record
         from the parent.
       • Move one level up the redirection history and query the nameserver for
         DNSKEY record
4: Call _verify_zsk() and verify the original DNSKEY response
5: Call _verify_rrset() and verify the RRSet from the original query's response
   using ZSK present in the original DNSKEY response.
```