DNSSEC Resolver - Implementation details

Mridul Ranjan, SBU ID - 111482003

DNSSEC requires support from both DNS name servers as well as DNS resolvers.We use dnspython library to single iterative DNS query. By default, 'A' record type is set for any domain. However, other types can also be passed to $mydig_sec$ script, which will invoke $my_sec_resolver.py$ with parameters.

1 Control Flow in the program

Doamin and category are passed from $mydig_sec$ script to main method of $my_sec_resolver.py$. Main method builds the pipeline of first resolving domain and category using DNSSEC protocol. This includes multiple steps which will be explained later. Once domain server is resolved,result(http://www.dnspython.org/docs/1.15.0/dns.message.Message-class.html) is passed to main method. Once this result is obtained, main delegates the work of printing the output as well as recursively resolving the DNS query to $format_result$ method.

2 Detailed Explanation

- Resolve domain category method - This method first tries to get list of name servers which which it can query to get answer. This part is delegated to $get_domain_servers$ method. Once list of name servers are available, this method makes a tcp query to each server with timeout of 1 second.TCP was used instead of UDP as discussed here in the forum : https://serverfault.com/questions/404840/when-do-dns-queries-use-tcp-instead-of-udp. If the name server gives response without exception, its passed back to main method else retried with other server.

- Get domain Server method - This method handles the most crucial part of DNSSEC protocol. It validates the chain of trust from root server and resolves sub-domain level by level. First of all, it queries the root servers (which are hard-coded) to get the top level domain server. Validation of root servers are done in $get_tld_server_details$ method. After getting the valid top level domain server for the requested domain it does following things in sequence. Along with list of parent domain servers it also gets DS record as well as algorithm used for hashing. Now it queries the child zone server for ZSK, RRSIG and RRset. If child server fails to provide ZSK and RRSIG records, it means DNSSEC is not supported by the domain. However, if ZSK and RRSIG are obtained from child domain, then they are validated to build the chain of trust. Algorithm field obtained from parent zone domain states whether sha1 or sha256 is used for hashing the keys RRset.Now ds record obtained from parent is validated with DNSKEY record in child zone after using the corresponding hashing algorithm. This is the key validation. Now, RRset validation is done. dnssec library provides validate method which validates the RRSIG obtained in child zone. If RRset is validated, that means key as well data both are validated in this parent-child zone pair. This process is repeated level by level to build the chain of trust. Once all sub-domains are traversed, list of trusted servers are send to the calling method.

- Get TLD server details method - As discussed over https://piazza.com/class/jcm92iy554oxx?cid=42, we hard code the DS record for root server to validate root servers. So first step in getting top level domain server for requested domain, is to validate the root server itself. We fetch the ZSK, RRSIG and RRset records from the root server. Now, we create DS record using ZSK and sha256. Now we validate the key by comparing the generated hash with hard coded root ds record. Once this validation is successful, we validate the RRset with RRSIG using the library validate method. Once the root server is validated, we obtain the ds and algo values from root server for the requested domain to the calling get_domain_server method which initiates the trust building by calling the sub-domain zone server and validating key and data.