LETTERKENNY INSTITUTE OF TECHNOLOGY

COVER SHEET

Lecturer's Name: Ruth Lennon

Assessment Title: External facing DNS with DNSSEC and telemetry/alerting for an LYIT subnetwork

Work to be submitted to: Blackboard

Date for submission of work: 10-07-2020

Place and date for submitting work: Letterkenny, 09-07-2020

To be completed by the Student

Student's Name: Anand George Francis (L00144568)

Class: MSc Cloud Technologies

Subject/Module: Placement Module

I confirm that the work submitted has been produced solely through my own efforts. Evidence for the practical element is stored in the repository **https://github.com/Anand-George-Francis/DNS-DNSSEC.**

Student's signature: Anand Date: 09-07-2020

Notes

Penalties: The total marks available for an assessment is reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. Assessment work received more than two weeks late will receive a mark of zero. [Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute's Assessment Regulations.]

Plagiarism: Presenting the ideas etc. of someone else without proper acknowledgement (see section L1 paragraph 8).

Cheating: The use of unauthorised material in a test, exam etc., unauthorised access to test matter, unauthorised collusion, dishonest behaviour in respect of assessments, and deliberate plagiarism (see section L1 paragraph 8).

Continuous Assessment: For students repeating an examination, marks awarded for continuous assessment, shall normally be carried forward from the original examination to the repeat examination.

External facing DNS with DNSSEC and telemetry or alerting for an LYIT Subnetwork

Anand George Francis
Department of computing
Letterkenny Institute of Technology
Letterkenny, Ireland
L00144568@student.lyit.ie

Abstract- A world without the internet is difficult to think of. All computers in the entire world are connected through the internet. Human access the internet online using DNS (Domain Name System) where a name matches to numbers. In general, human-readable names are converted into machine-readable numeric IP addresses. But there are certain fatal flows in the composition of DNS which leads it to exploited by fraud criminal users and serious of hijacking. These cyber-attacks to the DNS system provide the way for DNSSEC (DNS Security Extensions) which add a layer of security in DNS. The DNSSEC provides security to DNS servers to validate the information they are receiving. In this paper, the proposed system monitors and alert the fraud information that is received through LYIT subnetwork by implementing DNSSEC not only in domain name infrastructure but also in the DNS server. Furthermore, the entire configuration and implementation of these solutions are made through an automation using bash.

Keywords— Internet, DNS, DNSSEC, IP Address, BIND

I. INTRODUCTION

This Internet is a broad network that allows different networks around the world that are run by different organizations, universities, companies, and other individuals to communicate with each other. An outcome is a group of computers, routers, cables, data centers, satellites, repeaters, and WIFI devices that allow us to travel digital information around the world. Every computer connecting to another device on a network or anywhere on the world requires a unique IP address. Based on these IP addresses the network hardware can be identified. But they are difficult to remember by human beings since it is made up of set numbers or alphanumeric characters. There came into the use of DNS (Domain Name System)[1]. The DNS server eliminates the need of remembering this numeric address instead use of a name for easy identification. The DNS maps human-readable hostnames into machine-readable IP addresses before starting communication through the internet. Every individual uses DNS as a trustworthy mapping service for their day to day internet activities.

However, the progress of the web leads to a variety of security attacks. Like all other primitive internet protocols, DNS is also not equipped with this evaluation change. So that it leads to various attacks like the MITM attack, alteration of DNS data and application and system crashes. The DNS protocol itself considered as a secure communication protocol through which request-response queries are carried out, but a malevolent DNS response can lead to a phishing attack. Like Domain hijacking, DRDoS, DNS flood, Cache poisoning, DNS tunneling. DNS attacks happen mostly because of the user is not aware of DNS as a critical attack vector.

If regular DNS audit is not done by the user attackers use this opportunity. The common approach for handling these vulnerabilities is adding additional intelligent solutions but unfortunately, it is a temporary solution until the next attack. When DNS was implemented the internet is not huge that was today, so security is not concerned about the design. To diminish these attacks, several solutions were undertaken. The most commonly known is 'DNSSEC (DNS Security Extension)'[2]. Now a day, the key role of the internet is considered as the allocation of information in different data forms such as text, image, audio, and video to the requesting user. It seems simple to the users but unfortunately, it has been controlled through different infrastructure over the period such as DNS and DNSSEC. The DNSSEC provides an additional security feature to DNS. Based on public-key cryptography feature DNSSEC authenticates the DNS using digital signatures. By checking this digital signature on records, the user can verify the requested DNS records belong to an authoritative server and are not injected by a man in the middle attack. Like other protocols for security such as HTTPS (Hypertext Transfer Protocol Secure), DNSSEC add a layer for security over authorized answers. DNSSEC provides solutions for real-life problems without much in cooperation with cryptography.

The basic reason behind the production of DNSSEC infrastructure was based on the security threats along with fraud faced during domain naming scheme and pollution of cache examined and effect the infrastructure during Domain Name system were controlled with the help of DNSSEC. The plus features that were part of the infrastructure of DNSSEC are providing a platform for monitoring information in the real-time environment along with filtering of the harmful domain through network monitoring during naming scheme To solve this security design fault on DNS, in this paper, I describe the design and implementation of an automation script to monitor and alert the network. The objective of this report is to explain about the automation script that is created to monitor and alert LYIT (Letterkenny Institute of Technology) subnetwork by implementing features of DNSSEC on its external-facing DNS server. And, some background on DNS. The script divides into two phases. First phase, the BIND server and its related packages are installed and in the second phase domain will be added to DNS along with it DNSSEC is configured. The entire script is written to run only on either CentOS or RedHat systems.

II. THEORY OF DOMAIN NAME SYSTEM

Normally the user remembers the name of the website like google.com which is considered as the domain name in the field of the Internet. Whereas these domain names cannot be directly interpreted by web browsers because the information exists on the internet is interacted with the address of Internet Protocol (IP). By the implementation of DNS infrastructure, the web interaction between user and webpage is possible[3]. Due to existence of Millions of user on the internet, webpage interaction with the requesting user was addressed by DNS infrastructure while assigning a unique address to every device that is connected on the internet in a different type of memory address scheme such as IPv4 and IPv6 for the recognition of every machine uniquely or mapping process of a different device with each other. The Implementation of DNS infrastructure is in a structural form of hierarchy in which the Resolver act as the intermediary of both client and DNS server in such a way that for the client it takes queries and submits them for further processing upon it whereas on the DNS server end it respond by receiving queries from clients.

Thus, the Domain Name System is the mechanism that manages the interpretation of the naming scheme of the user device to an internet understandable pattern known as the Internet protocol. So, the transmission of resources that were acquired by users should be provided to the request users. The plus features that are part of the infrastructure of DNS are maintaining the address records of the end-user so at the time of information extraction or fetching the end-user must be known to system to shows the results of their request. Whereas due to the existence of vulnerabilities within the infrastructure of DNS were solved with the help of more secured known as 'DNS Security Extension'. Figure 1 shows the simplified operation of DNS.

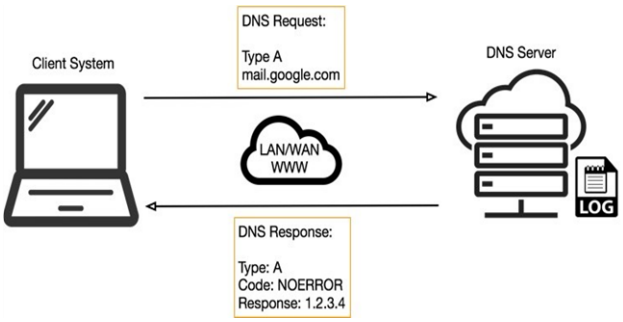

Fig 1. Simple DNS operation.
Source: Adapted from[4]

A. DNS components and its working

To understand how DNSSEC provides security extension to the DNS initially, we will start with the basic phrases of DNS. In General, the role of the DNS server is ,translate a Fully Qualified Domain Name into its corresponding IP addresses or translate the reverse. The request from a client can be termed as a 'query'. If the client asks for an IP address it is forward lookup query or the reverse, request for the hostname is called 'reverse lookup query'. The FQDN addresses the complete name of the domain that is required for a host as provided on the Internet. There has been the availability of two parts of FQDN which are the name of the domain and the name of the host. For example, myemail.lyitcollege.ie., in this id, myemail is the name of the host and lyitcollege.ie. is the name of the domain[5]. So, DNS is defined to be the database that is distributed, and which helps in the representation of a namespace. There has been the availability of different names that are formed in the structure of the tree which is being maintained with the help of the database globally. The client names and information are being widely available within the namespace and hence eliminating the fumbling attitude for the appropriate data. DNS is divided into different zone types. This portion of namespace is delegated to a specific company or organization. Whenever a zone is being hosted by the server of DNS, there has been the development of authority for grouping names for that zone[6]. The zones address a specific portion that is present in the namespace of DNS that is being controlled by these administrators. All types of records are being stored in every domain which is written in a text file that is being used for controlling the server of the DNS. With the help of that authority, the developed queries are being answered according to the names gathered in that zone. The IP addresses are being translated from their respective domain names with the help of DNS. Mail servers are being provided for accepting emails concerning their domain names.

DNS domain namespace is a hierarchical tree structure. DNS characterize domain namespace into various levels such as TLD, Second Level Domain, and subdomains. Every one of these levels can be a zone in DNS (www.example1.ie.). A DNS server can be a collection of zero, one, or more DNS zone. These systems consist of a primary zone (root server), which is represented as a dot (.) towards the end of the domain namespace. Each one of the DNS server software contains a list of root hints to identify these root servers. Root hints make use of the iterative queries. Whenever the resolution of a query is impossible, then queries are generated with the help of a database that is transferred to the DNS root server. The response is provided through a referral in which top-level domain addresses of the servers of DNS for the original query. The query that is generated by a local server is being answered for a domain for the second level. This continues until the query gets permission by FQDN, which will result in a negative or positive response[7]. Whenever it is impossible to solve queries for the server of local DNS, then the configuration is being provided to Forwarders that generate and transfer recursive queries amongst themselves present in the list. This will respond to negative or positive instead of providing a referral to the local server. The record is tracked with the help of a forwarder, which includes the process of referral[8]. Second is TLD zones, includes .com, country code (like ie), or org in the FQDN. The user viewing a part of FQDN (example1.ie) is a second-level domain that is treated as a separate zone and is operated by organizations or individuals. Usually, companies use their own name servers or given it to external providers. If the domain has a subdomain, then it can be in the same zone as well. Zones are characterized by authorized nameservers using zone files[9].

A zone file includes representation of zone and records for each domain within that zone. DNS Zone files can be further

classified into a Master File for describing a zone and a Cache File describe the contents of DNS cache. Zone file describes the DNS resource record (RR) in its each line. A record is consisting of different fields such as;

- Name: An alphanumeric value inherits from its count from its previous record.
- Time to live (TTL): Describes how much time the record will be kept in the local cache of DNS client.
- Record Data: Depending upon the record type this field contains additional elements of information.
- Record Class: Indicates the internet namespace
- Record Type: Indicates different DNS record types

DNS zone files include two mandatory fields. First, Global TTL, describes how long the records will keep in the local cache of DNS. Second, the Start of Authority record (SOA), this record is being utilized for the initiation of the records that are available in the authority of the server of DNS After these records, the zone file can consist of different RR[10]. Records of DNS are being utilized for pointing out subdomain or domain that will help represent IP address. The records are being stored in the specific zones of the DNS, which holds the information of the different resources that are being present. The most commonly used record types are;

- A Record: This record is being utilized for mapping of IP Address that is being available from the names that are present in the zones.
- CNAME: It is utilized for mapping a canonical name from an alias that is being present in the zones of the DNS server.
- MX Record: Provides the identification of the servers of the mails that are present in the zones of the DNS server.
- NS Record: This record is being utilized for the identification of the names that are available in the servers and which are being available for a specific zone.
- TXT: They are permitting the text that is required to be included in a record of the server of DNS.

Other than these described records, some records are less used such as; AFSDB, APL, CAA, DNSKEY, CDNSKEY, CERT, DCHID, DNAME, HIP, IPSECKEY, LOC, NAPTR, NSEC, RRSIG, RP, and SSHFP Record[11]. Figure 2 shows the structure of DNS.

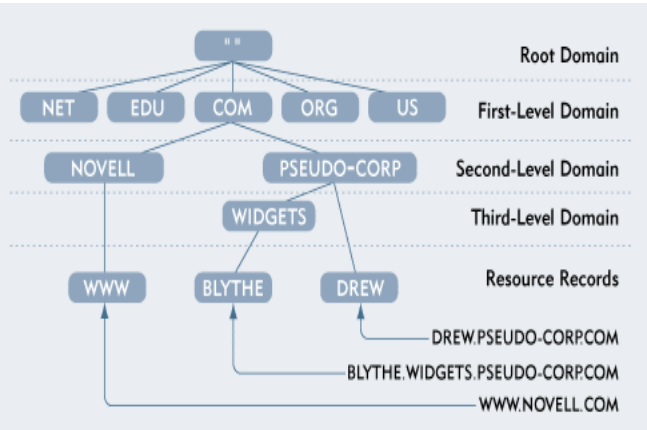

Fig 2. The hierarchical structure of DNS
Source: Adapted from [12]

DNS zone files are stored in Master or Slave nameservers. Master nameservers are also termed as 'primary nameservers' where changes on the files are made and they were truly authorized. The primary nameserver will have a read and write a copy of the DNS zone database. Whereas slave nameserver or secondary name servers will receive their DNS zone file from the master. Depends on the nameserver configuration, any nameserver can act as a master or slave for different zone[13]. One of the important features of DNS is zone transfer and it should be carried out securely [14]. Zone transfer safety can ensure by setting transfer only between a primary nameserver and authorized nameserver.

B. DNS vulnerabilities and security enhancement

After the deployment of DNS infrastructure, many internet attacks have occurred which shows the loose end of the infrastructure that highlights that the most crucial task over the Internet is compromised by a group of unauthorized people while breaching the network security. Through analysis different vulnerabilities were discovered in the existed infrastructure of DNS that compromises towards internet security was the denial of services, cache poisoning, tempering of data and discloser of information, discloser of mechanism which control the defect of the implemented infrastructure on the existed system and availability of server software to the unauthorized group of the user for creating a pool of changes in the retaining in the existing infrastructure[15]. Another drawback of the DNS infrastructure on the relationship of the slave-master principle which gives the wide opportunity to an unauthorized group of people to intrude on the master server and then performs phishing on the pages which are embedded within the domain of master server[16].

One of the best ways to eliminate these vulnerabilities and drawbacks of DNS is the adoption of DNSSEC protocol. Attackers found the different target to attack DNS, here are some of the major DNS attacks;

- DNS reflection: Whenever the server of DNS receives a query that is being generated by DNS, a response will be provided to the source of the generated query based on the IP address. Now, whenever an attacker tries to get an IP address from the query that is being generated by the victim, a response packet will be immediately sent to the victim from the server of the DNS. Now the response will be dropped by the victim by mistake assuming the fact that an intermediary server of DNS is generating this response. Now similar response packets are being received by the victim, and since it takes too much time in processing and downloading so, this results in network congestion.

- DNS amplification: Suppose a request that is formed by DNS leads to a packet if a more significant response, then it will be troublesome for the victim for revising the entire packet which leads to reflection attack amplification. The attacker sends many requests by taking the victim's name, which acts as a trap, and thereby other servers get trapped

and thereby affecting the infrastructures of other systems.

- DNS tunneling: In this process, the internal data gets stolen by the attacker that is present within a network, which in other words is termed as ‘data exfiltration’. In this process, initially, malware is being infected to a client of DNS by an attacker which leads to the development of a tunnel in the machine of an attacker with the help of a recursive server or DNS present in the network of the client and hence helping to bypass the network firewall. With the help of this generated tunnel HTTPs traffic can be passed in the form of a query-response of DNS.

- DNS hijacking: This is done by infecting the system with the help of Trojan or Malware which targets the client of DNS. There will be the modification of the TCP/IP settings of the clients and thereby sending queries of DNS to a rogue server of DNS. The attacker is controlling this process.

- Dos attack: Here an attacker targets a DNS server with the help of a single connection for flooding this server with the help of fake queries that are being provided to DNS which will ultimately lead to the total exhaustion of the server of the DNS.

- DDoS attack: In this attack, an attacker targets the server of DNS with the motive of flooding that server with the help of UDP requests. The requests that are being generated by UDP are being generated with the help of the Botnet network of compromised systems[17].

The lack of DNSSEC adoption together with other vulnerabilities causes DNS to become an important target for attackers. Other than DNSSEC there are other operations to protect against DNS attacks. Firstly, The DNS zone contains an operator to ensure the security of their servers. One of the simple methods to eliminate DDoS attacks is over-provisioning but it is difficult to handle volume-based attacks. Secondly, A handy tool called ‘Anycast routing’. It allows different servers to have a single IP address, if one of the DNS server shutdowns there will be another up and running. Third, DNS Firewall helps in providing high-level security that is being provided by filtering the traffic of DNS for providing suitable protection against the attacks on DNS and provides protection against the different types of queries of DNS which will further result in bad domains. This system helps in providing protection against the malware and thereby providing the functionality of Response Policy Zones. This process detects the attacks of DNS that are being generated by focusing on the signatures of those attacks and thereby defending them with the help of packets on the suitable matching of the signatures of those attacks. Specific domains get blocked, and DNS traffic gets monitored with the help of this system. And finally, the DNS resolver’s configuration can also provide some security to the end-users. They can provide features like content filtering, botnet protection[18].

C. Features Added to DNS by DNSSEC

DNSSEC provides two major features that are not available in the DNS protocol. First, Data Origin authentication, there is cryptographic verification of the available data that is performed by a resolver. The motive is to find the initial location or source of the data which gets acknowledged through verification. Second, Data Integrity protection, the resolver gets acknowledged about the fact that the modification of the available data was not performed while transitioning during the signing of that data by the owner of that zone with the help of the zone’s private key. A detailed explanation of DNSSEC is provided in the next section.

III. DOMAIN NAME SECURITY EXTENSION

When the Internet seemed companionable and trusting place there came to existence of DNS structure. The agreement itself provides required security against vindictive or answers that are forged. In case of flawless world, each DNS answer can be confirmed and trusted at the time of DNSSEC being fully established. DNS data will not be hided or encrypted by DNSSEC as it does not give way for assured tunnel. Public Key Infrastructure (PKI) is existing structure which operates it independently. SSL certificates or any shared secrets are not required here. It was structured with backwards similarity in mind and that was established without collision to "old" insecure named domain

Exploration of DNS through different internet threats was addressed as the implementation of DNSSEC infrastructure, which embedded the DNS infrastructure while encrypting the topmost layer for the purpose of authentication in the existing infrastructure for controlling the intruder phishing or intrusion attack on the servers. Whereas the unique features of the infrastructure reside in the signature of cryptographic (Public Key encryption mechanism) implementation on the address of name domain from the user and holding the information in DNS records while encoding in digital signals for securing the whole process. By this factor, the system becomes effective enough to verify the effectiveness of the user domain along with securing the user information form any sort of disclose. Due to the implementation of the data signal in the infrastructure, the data integrity and authentication of the origin of data were achieved with the help of DNSSEC infrastructure[19]

.

A. Why domain name security extenstion

With the help of the DNS database, a digital signature that is already pre-generated was implemented on every individual information to provide authentication of originality along with data integrity while using the DNSSEC infrastructure. Due to these reasons, Domain Name System Security Extension ensures that the infrastructure deducts the attack of MITM intruder who is supposed to create colonial traffic along with eavesdrop at any service. The reason behind this adaptation of this mechanism is that it utilizes the existing features of the Domain Name System along with the implementation of a lack of that infrastructure. In other words, to provide services through DNSSEC the security factor is applicable on DNS transactions at both endpoints during adaptation along with the security is considered during

maintenance and deployment of the servers. DNSSEC is created to secure internet resolvers form molded DNS data. It is a group of extension to Domain name system clients or resolvers[20];

- Authenticate origin of DNS data
- Provides data integrity but not confidentiality
- Verifies existence

All response queries are digitally signed by in DNSSEC. Using this signature, a DNS client will able to check whether the information is similar to authoritative server. DNSSEC also provides protection for certificates stored in the CERT records as well.

B. Working of DNSSEC

In general, DNSSEC will digitally sign the DNS records in the authoritative server using public-key cryptography. For these functions a number of RR types are introduced;

- Resource Record Signature: A signature of DNSSEC that is being provided for a set of records having a similar type of name is developed with RRset Signature.
- DNS Public Key: DNSKEY contains a "public" key that is used for verification of DNSSEC signatures into records of RRSIG with the help of resolvers.
- Delegation Signer: It has the digital information of Signature and is further used for signing the key of DNSSEC for a particular zone.
- Next Secure: NSEC points to succeeding record name for the desired zone also displays the record types that are available for the record name.

The DNSSEC also introduced two new types of DNS header flags such as;

- Authentic Data: This bit specify that all data are included in the response answer as well as the authority section is validated by the server using their polices[21]
- Checking Disabled: This bit is used to return when RRset is not verified.

While DNSSEC where used, each response query of DNS lookup will contain an additional RRSIG record along with other requested record types. The RRSIG is the digital signature of the response RRset. This digital signature is verified by detecting the correct public key inside the DNSKEY record. The DNSKEY is cryptographic public key consist of two roles;

- Key Signing Key: It is defined to be the key for signing the DNSKEY that is required for validation purposes, which are also a combination of both private key and public key.
- Zone Signing Key: It is defined to be the key for encryption that helps in signing a zone basically with the help of a combination of a private and public key for storing the data in a certificate and it will sign all other records in the domain.

From this result, the DNS resolver can determine whether the received response is true or some error. The valid DNSKEY is found using authentications chains that begin with a public key for trust anchors. The Trust anchors help in the validation of data that is being provided to a signed zone with the help of a public cryptographic key. A trust anchor should configure on all non-authoritative servers who can validate the data[22]. This public key then further used to validate the DS records. DS record in the DNS zone can then verify the DNSKEY in its subdomain, which includes another DS record to validate further child domain. The global components and records of DNSSEC are shown in the below figure 3.

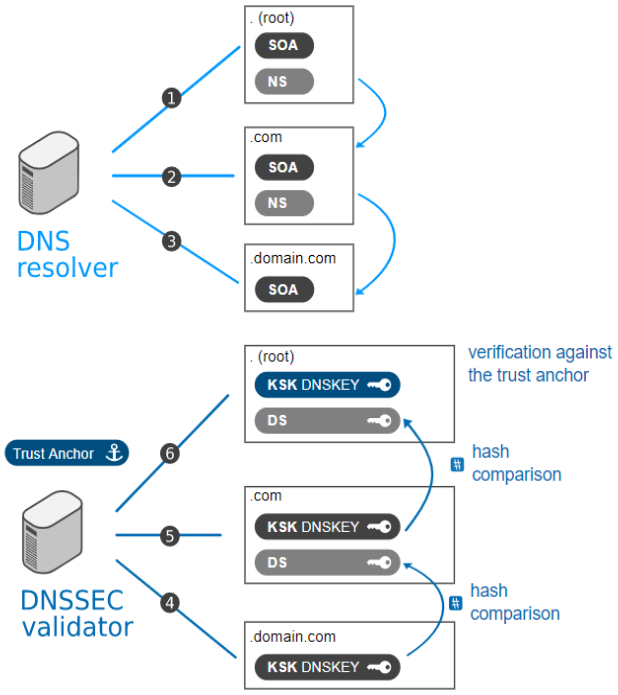

Fig 3. Components and records of DNSSEC
Source: Adapted from[23]

C. DNSSEC authentication mechanisim

Initially, the local authorized server gives the RR set hash value by a hash function and encrypt this value by its private key. This encryption output a RR with digital signature. When an authoritative accepts queries from a complete DNS resolution, it will respond back to it RRSIG and resource record. Otherwise, when a DNS accept response from a corresponding authoritative server, it will evaluate hash value using resource records hash function. Also, it will erase RRSIG value received using DNSKEY. The public key of DNSKEY is able read the private key of its parent domain. If the evaluated hash value is exactly same as RRSIG, then DNS will be a resolution resolver. Finally, When those two hash values are matched, the DNSSEC authentication will be successful. Figure 4 shows the authentication mechanism of DNSSEC.

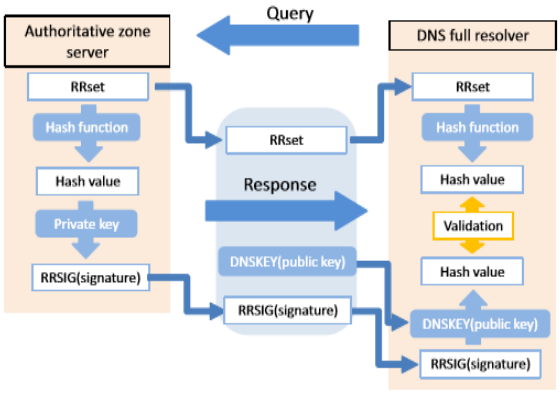

Fig 4. The authentication mechanism of DNSSEC
Source: Adapted from[24]

D. Implementation of DNSSEC

The implementation process of Domain Name System Security Extension is not far complicated but generally splinted into multi-process at each step while depending upon the registrar of domain's, site extension top-level domain and the configuration of the name server to be managed. The implementation process can be defined in the following way such as: extension of security based on DNS should be supportable by TLD, gathering the requirement related or specific to DNSSEC, while relation the DNS to your Zone request for generation of ZSK and KSK for a process, cryptographic keys were utilized for the encoding of DNS before the DS generation and declaration. Whereas the most important part of the implementation involved the self-host on which the DS record should be imported through the user domain. Further on an exceptional case scenario was designed for the encountering of Chain of trust during the DNSSEC implementation process otherwise it will affect the DNS server to perform better and data integrity and security might be compromised because of the factor. This implementation is capable of the generalized setup of Domain Name System Security Extension whereas the name server self-managed requires more configuration then this process. Whereas DNSSEC infrastructure monitoring was carried out in ways such as the construction of a nation-level trust chain, records of local recursive domain servers, and construction of DNSSEC monitoring.

IV. AUTOMATION SCRIPT FOR EXTERNAL FACING DNS WITH DNSSEC AND TELEMETRY/ALERTING FOR PROPOSED SUBNETWORK

The automation script is created for an external-facing DNS with DNSSEC and having features of telemetry or alerting entirely for LYIT subnetwork. The implementation of this proposed automation script is not too much complicated, but it splinted into a different process. To provide a meaningful result, we must define; what is BIND[25].
BIND is the most popular open-source software for the implementation of DNS. They translate the domain into IP addresses and the reverse translation. BIND is usually available in Linux servers. The BIND (Berkeley Internet Name Domain) is a frequently used DNS software because it was using ETF standards. It can run on a Windows machine, but it still requires certain updating. BIND carryout both two roles of the DNS server, act as an authority's server for a domain and recursive resolver for the network. BIND provides different features and capabilities for DNS, and for our proposed solution it is DNSSEC[26]. It will cryptographically sign the authorized data and verify the data on the cache server. BIND provides support for recent iterations of DNSSEC such as elliptic curve cryptography.

A. Design and Implementation

The general requirement specification includes[27];

- Hardware Requirement: Traditionally hardware requirement for the DNS is quite modest. However now, the BIND feature of DNSSEC is little CPU intensive. So, an organization that is using this feature should use large systems.

- CPU Requirement: CPU requirement ranges from the i386 class machine, for handling thousands of queries per second.

- Memory Requirement: Enough server memory is required to hold both zones and cache. There are different options to limit the memory used by the cache.
- Operating Systems: BIND can compile and run different UNIX/LINUX based systems.

For the implementation of the script, created a new virtual machine with *CentOS7* and configured with a memory of 2GB, 2 processors, and a 20GB of hard disk.
The script is entirely designed only for running on RHEL based systems such as RedHat, CentOS, or Fedora using Bash scripting. And the implementation of this script is narrow and specific.
In the first phase, the BIND server and its related packages will be installed. And in the last phase, the domain will be added to DNS along with it DNSSEC is configured.

This script can run only as a *root* or *sudo* user. In the first phase, the BIND server installation is done.
For this, the first argument is given as parameter ***install***, for installing BIND in the supported machine. The configuration for the BIND server is saved under the file */etc/named.conf* [28]. The script will check whether this file is already existing in the machine or not, if it is existing it will not further install the package and generate error alert. If the file is not found then, the script will check whether the machine is running on RHEL or Fedora-based system. It is because the script is entirely written for run on RedHat and CentOS machine. If these conditions are satisfied, then the script will instruct to install BIND based on the type of operating type. Figure 5 shows this corresponding part of the script.

Fig 5. Package installation of BIND

In general, for configuring the BIND server, initially, the script tells BIND to listen on which port and network interface. This is done through the command *listen-on port 53 {127.0.0.1; ip-address;};* where port 53 is default port number to accept client queries. The 127.0.0.1 loopback permits request that are raised from the localhost. If this field is omitted all public interfaced will be permitted. This is the case for IPv4 client queries. For dealing with IPv6 client queries we use the command *listen-on-v6 port 53 {any; };* . And finally, for defining the network through which the client can raise DNS queries is enabled using the command *allow-query {127.0.0.1; net; };* [29].

But in our script the above mentioned three configurations are replaced their value as *any* in the conf file, this is because our DNS server is external facing. This will provide any client to request to any IP in the DNS server. Figure 6 shows this corresponding part of the script.

Fig 6. The basic configuration of BIND

After the packages are installed, the BIND server should start and enabled along with haveged package then only services will be get started. Here it uses *haveged***,** is pseudo number generator, and nourish Linux systems. They use Hardware Volatile Entropy Gathering and Expansion algorithm to reduce the workload of the system and help for key generation. Systems such as virtual machines require more amount of time to generate keys. If haveged is not adopted the script will consume more time to run and it will take more time to generate keys[30].

For starting the BIND named server the script uses the command *service named start* or *systemctl start named*. To enable BIND service, use the command depending on the system i.e. *chkconfig named on* or *systemctl enable named*. And for starting haveged use *service haveged start* or *systemctl start haveged* and for enabling use *chkconfig haveged on* or *systemctl enable haveged*. Figure 7 illustrates the start and enabling of named and haveged.

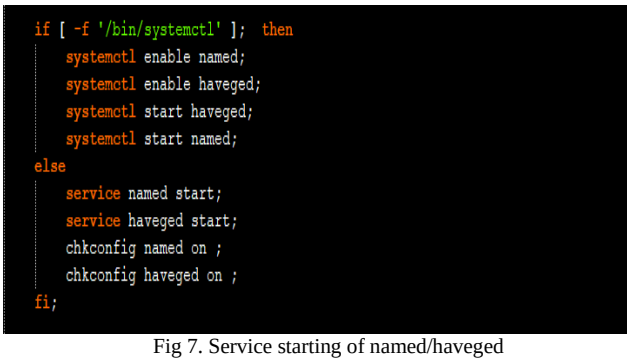

Fig 7. Service starting of named/haveged

Once the services are started the script will recheck whether the BIND is successfully installed or not. Today every *named.conf* is enabled with different DNS security statements such as *dnssec-enable*, *dnssec-validation* by default otherwise we must deploy these DNS security extensions[31]. So, the script will check whether the *named.conf* is enabled with enable option or not. If it is not enabled, the script will add *dnssec-enable* to the 40th line of *named.conf*. Same as the case of *dnnsec-validation*. The dnssec-enable shows that it is used with a secure DNS service such as DNSSEC. Whereas *dnssec-validation* will validate the replies received from DNSSEC signed zones.

But the DNS mechanism such as *dnssec-lookaside* is not included as default, so we must add *dnssec-lookaside* in 42nd line of *named.conf* file for creating a group of domains known 'DLV domain' to provide a secure access point for different zones that are not their children[32]. Also maps the key directory to the *named.conf*. Figure 8 shows this corresponding part of the script.

Fig 8. Deployment of DNS security extenstion

The script uses *firewalld* mechanism for the security. In general, a firewall can be termed as a security mechanism to monitor the packets that are flowing through the traffic based on some predetermined rules. Usually *firewalld* is a default option by CentOS but it won't be active. So, in the script once the *firewalld* status is checked it will add certain rules as per the requirement. Rules can be designed as either permanent or temporary. Initially, by the *firewalld*, the connections made from outside will be disabled so for our requirement we need to add certain rules. For these reasons, the script will certain rules such as; adding DNS service using the command *firewall-cmd --permanent --add-service=dns*. DNS uses port 53 as its default port which is working on UDP and it should open using the command *firewall-cmd --permanent --add-port=53/udp;*. If the response is more than 512 bytes then the

request is sent through *tcp port 53*. Whenever a rule is updated, *firewalld* is needed to be restarted otherwise rules will not activate [33]. Figure 9 shows the corresponding firewall configuration on the script.

```
if [ -d /etc/firewalld/zones/ ] && [[ $(systemctl is-active firewalld) = "active" ]]; then
 firewall-cmd --permanent --add-service=dns;
 firewall-cmd --permanent --add-port=53/tcp;
 firewall-cmd --permanent --add-port=53/udp;
 firewall-cmd --reload;
fi;
```

Fig 9. Firewalld configuration

Before the zone creation, the script consists of certain best practices. To make *named.conf* upright, the script creates a separate file called *named.conf.local* in which DNS zones can be defined. And for making it called at the time of service start, it will be included under *named.conf* file. Also, the ownership of the file is changed otherwise BIND service is not able to read it. After completing all these basic configurations, the system is restarted so that changes will be updated, and the machine will ready for the next phase. Figure 10 shows this corresponding part of the script.

```
echo 'include "/etc/named.conf.local";' >> /etc/named.conf;
touch /etc/named.conf.local;
chown root:named /etc/named.conf.local;

if [ -f '/bin/systemctl' ];  then
    systemctl restart named;
else
    service named restart;
fi;
```

Fig 10. Creation of separate file for a zone and changing its ownership

After the BIND installation process is done script will move to the second phase.

At this point, we need to add required domain name to the BIND server and for that, choose the argument ***add-domain***. Also, check another parameter to verify the domain name. Here script will expect one domain name if it's not entered or an invalid item is given then an alert message will be generated. Once the desired parameter is given script will do an additional check for the existence of *named.conf* file. If this check is successfully completed, the script will declare two variables for the zone such as the serial number and domain name[34].

Every zone will have a serial number. Whenever an alteration is made on the zone, the serial number also needs to be updated otherwise changes made in the zone will not be affected universally. The default format for DNS serial number is *serial=$(date +%Y%m%d) 00;*. Next, the script will ask to enter the corresponding IP address for the given domain and it will evaluate the authenticity of given IP. Figure 11 shows this corresponding part of the script.

```
elif [[ $1 = 'add-domain' ]]; then
    if [ -z "$2" ]; then
        echo -en "\n Incorrect usage, domain name expected : \n\t $0 add-domain domainname.com \n\n\n\n";
        exit 3;
    elif [ -f /etc/named.conf ]; then
        serial=$(date +%Y%m%d)00;
        domain="$2";
        echo -en "\n\t\tEnter Target host IP pf the domain : "; read ip ;
        while [ true ]
        do
            if [[ $ip =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$ ]]; then
                break;
            else
                echo -en "\n\n\tSeem not a valid IP, try again : "; read ip ;
            fi;
        done
```

Fig 11. Validate the domain name and IP

Next, the script will check whether the previously created domain is existing in the *named.conf.local* file or not. If the domain is existing on the file, then the script will exit otherwise a new zone file will be created under */var/named* consisting of given specific format. Figure 12 shows the zone file format with domain settings and RR.

```
            check=$(grep -w $domain /etc/named.conf.local | head -1 |cut -d '"' -f2);
            if [ -z "$check" ]; then

                cat << EOF > /var/named/$domain
\$TTL 1h
@       IN      SOA     $domain.    root.$domain. (
                        $serial       ; Serial YYYYMMDDnn
                        24h             ; Refresh
                        2h              ; Retry
                        28d             ; Expire
                        2d              ; Minimum TTL
                        )

;Name Servers
@       IN      NS              ns1.$domain.
@       IN      NS              ns2.$domain.

;Mail Servers
@       IN      MX      0       mail.$domain.

@       IN      A        $ip

;Other Servers
ns1  IN      A              $ip
ns2  IN      A              $ip

;Canonical Names
www     IN      CNAME          $domain.
mail    IN      CNAME          $domain.
EOF
```

Fig 12. Zone file format with domain and RRset settings

Once it's done, get into the previously created zone file */var/named* and generate keys for the previously created domain. DNSSEC operation requires two pairs of keys. Where *zone signing key* will be created for the domain with different attributes and a size of 2048 *ZSK=$(dnssec-keygen -a NSEC3RSASHA1 -b 2048 -n ZONE $domain);* and a *key signing key* will created a size of 4096 *KSK=$(dnssec-keygen -f KSK -a NSEC3RSASHA1 -b 4096 -n ZONE $domain);*. Those two keys will be appened to the corresponding zone file.

Finally, sign the zone using these generated keys[35] by the command *dnssec-signzone* as the domain name. The *dnssec-signzone* is a BIND tool for signing the zone and the given

parameter *O* defines the origin of the zone[36]. The signing is done only after fetching and evaluating these keys from the *key-directory*. And thus, the zone will be signed using the script and stored under */var/named* as a new signed file.

For the configuration, we have to append it along with *named.conf.local* file. And then create a zone file for the previously created domain. And add different parameters such as; *type master* it is because our server is a master server and signed file name for a name server to correctly pointing to the signed zone file, DNSSEC key location, etc. Figure 13 shows the corresponding part of this script.

```
        cd  /var/named/ ;
        ZSK=$(dnssec-keygen -a NSEC3RSASHA1 -b 2048 -n ZONE $domain );
        KSK=$(dnssec-keygen -f KSK -a NSEC3RSASHA1 -b 4096 -n ZONE $domain);
        cat $ZSK.key  $KSK.key >> /var/named/$domain ;
        cd /var/named/ ;
        dnssec-signzone -A -3 $(head -c 1000 /dev/random | sha1sum | cut -b 1-16) -N INCREMENT -o $domain -t $domain ;


        cat << EOF >> /etc/named.conf.local
zone "$domain" IN {

        type master;

        file "/var/named/$domain.signed";

        allow-update { none; };
        # DNSSEC keys Location
          key-directory "/var/named";

        # Publish and Activate DNSSEC keys
          auto-dnssec maintain;

        # Use Inline Signing
          inline-signing yes;

};
EOF
```

Fig 13. Signing of zone

The *user* and *group* ownership for the */var/named* need to change into *named* for resolving any feature issues related to ownership and restart the BIND to reflect the change. Finally loads the keys using the command *rndc loadkeys* for the corresponding domain[37]. And thus, the domain will be added to the DNS and DNSSEC is configured using the proposed script. Figure 14 shows this corresponding part of the script.

```
        chown named:named /var/named/ -R;
        if [ -f '/bin/systemctl' ];  then
            systemctl restart named;
        else
            service named restart;
        fi;

        rndc loadkeys $domain ;

        if [ -f '/bin/systemctl' ];  then
            systemctl restart named;
        else
            service named restart;
        fi
        echo -e "\n\n$domain is added to the DNS and configured DNSSEC : Success\n\n";

        else
            echo "Domain Already existing in the DNS server, please remove it and add again";
        fi;
    else
        echo -e "\n\n\nbind install not found, please install it first using $0 install \n\n\n";
    fi;
else
    echo -en "\n Incorrect usage, arguments expected :   \n\n\t $0 install   : for install bind\n\t $0 add-domain domainname.com \n\n\n\n";
    exit 3;
fi;
else
    echo -e "\n\n\n\t\tScript must be executed as root / as sudo user\n\n";
fi;
```

Fig 14. Loading the keys and restart to enable the updation.

B. *Evaluation and Result*

The created bash script is written to run only CentOS or RedHat systems. Currently, the script is not implemented on LYIT subnetwork, so we perform testing on the localhost. The entire operation can be done either as a root user or as a sudo user. Once login as root, the script starts to execute and ask for the desired operation to perform. Figure 14 shows the two options that are implemented with the script. The two options are;

- BIND server and its related packages are installed.
- The domain will be added to DNS along with DNSSEC is configured.

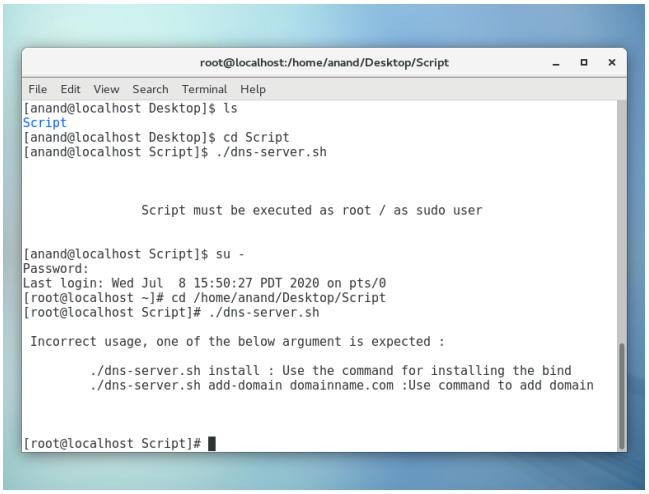

Fig 14. Two operations performed by the script

Source code available at https://github.com/Anand-George-Francis/DNS-DNSSEC. To install the BIND, choose the first parameter '*install*'. The script will then check whether the system is already installed with BIND or not. And check the operating system of the machine. If these conditions are satisfied, the BIND named server will be installed. As mentioned in the above *Section (A)*, all the necessary configuration settings for the BIND server, like dnssec security statements, firewalld are successfully installed and updated. Figure 15 shows the successful installation of the BIND named server.

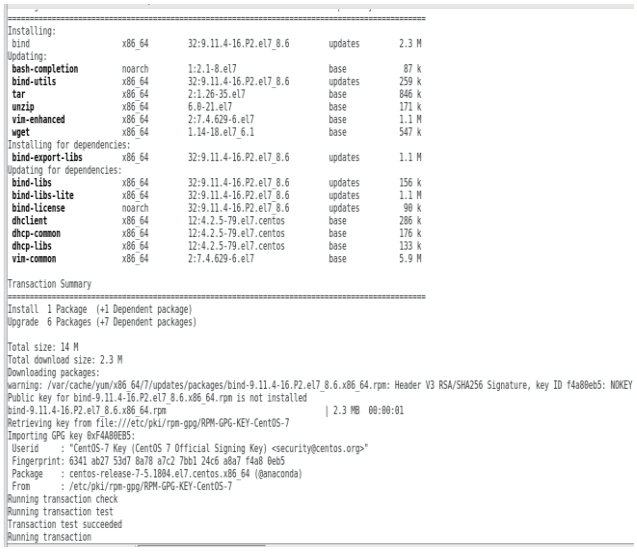

Fig 15. Installation of the BIND server.

If the condition is not satisfied, the script will generate an alert message as below.

```
[root@localhost Script]# ./dns-server.sh install


 Found bind files, please remove them and restart this script again


[root@localhost Script]#
```

Fig 16. The alert message generated when bind is already existing.

Once the BIND is successfully installed, choose the second option *'add-domain* 'to add the domains. Enter the domain name for the host and give its target IP address. If one of these arguments is not given, the system will generate an error message as below.

```
[root@localhost Script]# ./dns-server.sh add-domain lyit.example.com

              Enter Target host IP pf the domain :


      Seem not a valid IP, try again :
```

Fig 17. The alert message generated for invalid input.

For testing; the domain name is given as 'lyit.example.com' and give the IP of our localhost as 192.168.220.134. If it was in the real scenario; we must give the domain name of LYIT and IP address of its host. Here, the two keys such as *ZSK* and *KSK* keys will be generated and are verified by *NSEC3RASHA1* algorithm. And then the zone will be signed by the owner using the generated keys. Figure 18 shows the successful configuration of DNS with DNSSEC

```
      Seem not a valid IP, try again : 192.168.220.134
Generating key pair.............................+++ ..................................................
Generating key pair.....................................................................................
.................++
Verifying the zone using the following algorithms: NSEC3RSASHA1.
Zone fully signed:
Algorithm: NSEC3RSASHA1: KSKs: 1 active, 0 stand-by, 0 revoked
                         ZSKs: 1 active, 0 stand-by, 0 revoked
lyit.example.com.signed
Signatures generated:                      16
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.030
Signatures per second:                530.697
Runtime in seconds:                     0.136


lyit.example.com is added to the DNS and configured DNSSEC : Success


[root@localhost Script]#
```

Fig 18. configuration of DNS with DNSSEC

To verify the above result, opened the key-directory '*/var/named'*. We could see that the directory successfully contains the domain name, four keys (public and private), and signed zone file. Figure 19 shows the contents inside the /var/named file.

```
[root@localhost Script]# cd /var/named
[root@localhost named]# ll
total 76
drwxrwx---. 2 named named    23 Jul  8 16:32 data
-rw-r--r--. 1 named named   173 Jul  8 16:52 dsset-lyit.example.com.
drwxrwx---. 2 named named    60 Jul  8 16:53 dynamic
-rw-r--r--. 1 named named   960 Jul  8 16:52 Klyit.example.com.+007+02394.key
-rw-------. 1 named named  3319 Jul  8 16:52 Klyit.example.com.+007+02394.private
-rw-r--r--. 1 named named   616 Jul  8 16:52 Klyit.example.com.+007+54749.key
-rw-------. 1 named named  1779 Jul  8 16:52 Klyit.example.com.+007+54749.private
-rw-r--r--. 1 named named  2401 Jul  8 16:52 lyit.example.com
-rw-r--r--. 1 named named 11184 Jul  8 16:52 lyit.example.com.signed
-rw-r--r--. 1 named named   512 Jul  8 16:52 lyit.example.com.signed.jbk
-rw-r--r--. 1 named named  7382 Jul  8 16:52 lyit.example.com.signed.signed
-rw-r--r--. 1 named named 12007 Jul  8 16:52 lyit.example.com.signed.signed.jnl
-rw-r-----. 1 named named  2253 Apr  5  2018 named.ca
-rw-r-----. 1 named named   152 Dec 15  2009 named.empty
-rw-r-----. 1 named named   152 Jun 21  2007 named.localhost
-rw-r-----. 1 named named   168 Dec 15  2009 named.loopback
```

Fig 19. Files inside the /var/named.

Also verified the newly signed zone file by viewing the directory '*/etc/named.conf.loca'l.*

```
[root@localhost named]# cat /etc/named.conf.local
zone "lyit.example.com" IN {

        type master;

        file "/var/named/lyit.example.com.signed";

        allow-update { none; };
        # DNSSEC keys Location
           key-directory "/var/named";

       # Publish and Activate DNSSEC keys
           auto-dnssec maintain;

       # Use Inline Signing
           inline-signing yes;


};
```

Fig 20. Details inside the zone file.

For further validation of the result, use the dig command '*dig lyit.example.com @localhost'* for querying the created DNS server. It will display all the details included in the created domain such as records and air code. Figure 21 shows the records and air code of the domain.

Finally, the entire script can be verified using another dig command to generate DNSKEY. If the script successfully

configured DNS with DNSSEC, then the dig command '*dig DNSKEY lyit.example.com @localhost'* will output DNSKEY generated from the DNS server.

```
[root@localhost named]# dig lyit.example.com @localhost

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-16.P2.el7_8.6 <<>> lyit.example.com @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65361
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;lyit.example.com.              IN      A

;; ANSWER SECTION:
lyit.example.com.       3600    IN      A       192.168.220.134

;; AUTHORITY SECTION:
lyit.example.com.       3600    IN      NS      ns2.lyit.example.com.
lyit.example.com.       3600    IN      NS      ns1.lyit.example.com.

;; ADDITIONAL SECTION:
ns1.lyit.example.com.   3600    IN      A       192.168.220.134
ns2.lyit.example.com.   3600    IN      A       192.168.220.134

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed Jul 08 18:25:49 PDT 2020
;; MSG SIZE  rcvd: 129
```

Fig 21. The records and air code of the domain.

```
; <<>> DiG 9.11.4-P2-RedHat-9.11.4-16.P2.el7_8.6 <<>> DNSKEY lyit.example.com @localho
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45282
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;lyit.example.com.              IN      DNSKEY

;; ANSWER SECTION:
lyit.example.com.       3600    IN      DNSKEY  256 3 7 AwEAAfv9xpZBavdL8WXF5RCtxgyRBd
HhxANcurj7W0IeOHJdfEKFyhnYMgYKIoI6MGO XT3SjxaXbfs7/uUMpdE0hMjYqCN37JFKwnpA3Iz1KoITxrcP
Z1cw/rCmgXKhhau2XLM Qcf/Bgqe8zU=
lyit.example.com.       3600    IN      DNSKEY  257 3 7 AwEAAbvbqYnHa54W39HauU0D6deFRj
6T/L6/9PZKJQkJgAuPuw2uijFM2qDrT0fT095 sl528lZ1u/+1Ui28x0n6BGrxR/0ItS9JP2FdfJnGkl9W7c18
LpuTSpal08JK77trOyq Qf35CKv9MtNQqRJxKkGoSsy7vFqMNrmcxs1ymUaecmmry/N1Ymm2iQ5a pAlPOK4zh
7 z8Egijh5856CGSgPhGSfibE9eS1jm0BwvDHxVJ69P/3qbbeLCnsKJLxw v+fzcUU4y0dZqlToELz5UvTrEYw

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed Jul 08 18:22:43 PDT 2020
;; MSG SIZE  rcvd: 853
```

Fig 22. Verification of DNSKEY trust anchors

V. Conclusion

DNS is developed at the time of the internet is very small and security is not concerned. As a result, when a query is sent from a recursive resolver to a authoritative server, there will be no way to validate the authenticity of received query response. The one way to validate the authenticity is, verify the incoming IP address and if it is the same as the original query then the resolver needs to rely on it. But this option is not a secure because IP address of the source can easily alter by attackers. And it will cause massive attacks on the system. To resolve this issue DNSSEC is implemented. The DNSSEC provides strong authentication using digital signatures. By DNSSEC instead of signing just the queries, the entire DNS data itself will be signed by the owner. It is concluded that DNS gets new features of security with the help of DNSSEC. Now, for every organization or company, the major concern is its data security. The main objective of this paper is to create an automation script for an external-facing DNS with DNSSEC and having features of telemetry or alerting for LYIT subnetwork. The above result sample shows how easily this objective can be achieved using an automation bash script. The script is proposed and tested in a virtualized CentOS environment. And thus, it is concluded that we can create an automation script to establish DNSSEC over a DNS server in LYIT subnetwork as well as continuous monitoring and alerting options for it.

Acknowledgment

This paper is supported by many people from LYIT to accomplish the task.

References

[1] "What is DNS? – Introduction to DNS - AWS," *Amazon Web Services, Inc.* https://aws.amazon.com/route53/what-is-dns/ (accessed Apr. 25, 2020).

[2] "Domain Name System Security Extensions (DNSSEC)," Oct. 24, 2014. www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzab6/dnssec.htm (accessed May 10, 2020).

[3] 06 Apr 2017 David BothFeed 375up 5 comments, "Introduction to the Domain Name System (DNS)," *Opensource.com*. https://opensource.com/article/17/4/introduction-domain-name-system-dns (accessed May 10, 2020).

[4] "DNS Tunneling: how DNS can be (ab)used by malicious actors," *Unit42*, Mar. 15, 2019. https://unit42.paloaltonetworks.com/dns-tunneling-how-dns-can-be-abused-by-malicious-actors/ (accessed Jul. 05, 2020).

[5] "Fully Qualified Domain - an overview | ScienceDirect Topics." https://www.sciencedirect.com/topics/computer-science/fully-qualified-domain (accessed Jul. 06, 2020).

[6] N.-I. D. and T. Management, "DNS Zones Explained," *NS1*. https://ns1.com/resources/dns-zones-explained (accessed Jul. 06, 2020).

[7] R. Chandramouli and S. Rose, "Secure Domain Name System (DNS) Deployment Guide," National Institute of Standards and Technology, NIST SP 800-81-2, Sep. 2013. doi: 10.6028/NIST.SP.800-81-2.

[8] D. Mosley, "Root Hints and Forwarders - Designing Infrastructure Windows Server 2003," *Windows Server Brain*, Oct. 04, 2019. https://www.serverbrain.org/designing-infrastructure-2003/root-hints-and-forwarders.html (accessed Jul. 06, 2020).

[9] steve, "DNS Zones and Zone Files Explained." http://www.steves-internet-guide.com/dns-zones-explained/ (accessed Jul. 06, 2020).

[10] "DNS Records," *Cloudflare*. https://www.cloudflare.com/learning/dns/dns-records/ (accessed Jul. 05, 2020).

[11] "Chapter 4. introduction to DNS." http://linux-training.be/servers/ch04.html (accessed Jul. 05, 2020).

[12] "Snapshot." Accessed: Jul. 05, 2020. [Online]. Available: https://steemit.com/steemiteducation/@azizali/introducing-dns-server-technology.

[13] "Nameserver Zones - CentOS - Linux." http://underpop.online.fr/l/linux/en/centos/s2-bind-introduction-zones.htm (accessed Jul. 05, 2020).

[14] "DNS zone file and zone transfer," *ClouDNS Blog*, Oct. 03, 2018. https://www.cloudns.net/blog/zone-transfer-zone-file-domain-namespace/ (accessed Jul. 06, 2020).

[15] M. A. Hussain, H. Jin, Z. A. Hussien, Z. A. Abduljabbar, S. H. Abbdal, and A. Ibrahim, "DNS Protection against Spoofing and Poisoning Attacks," in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, Jul. 2016, pp. 1308–1312, doi: 10.1109/ICISCE.2016.279.

[16] M. H. Jalalzai, W. B. Shahid, and M. M. W. Iqbal, "DNS security challenges and best practices to deploy secure DNS with digital signatures," in *2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Jan. 2015, pp. 280–285, doi: 10.1109/IBCAST.2015.7058517.

[17] "SecurityTrails | The Most Popular Types of DNS Attacks." https://securitytrails.com/blog/most-popular-types-dns-attacks (accessed Apr. 25, 2020).

[18] "DNS Security," *Cloudflare*. https://www.cloudflare.com/learning/dns/dns-security/ (accessed Jul. 06, 2020).

[19] "DNSSEC – What Is It and Why Is It Important? - ICANN." https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en (accessed Jul. 08, 2020).

[20] "How Does DNSSEC Work?," *Bluecat Networks*, May 01, 2019. https://bluecatnetworks.com/blog/breaking-down-dnssec-how-does-it-work/ (accessed Jul. 08, 2020).

[21] B. Wellington <brian.wellington@nominum.com>, "Redefinition of DNS Authenticated Data (AD) bit." https://tools.ietf.org/html/rfc3655 (accessed Jul. 09, 2020).

[22] Archiveddocs, "Trust Anchors." https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn593672(v=ws.11) (accessed Jul. 09, 2020).

[23] "What is DNSSEC?," *EfficientIP*. https://www.efficientip.com/what-is-dnssec/ (accessed Jul. 09, 2020).

[24] Y. Jin, K. Kakoi, N. Yamai, N. Kitagawa, and M. Tomoishi, "A Client Based DNSSEC Validation System with Adaptive Alert Mechanism Considering Minimal Client Timeout," *IEICE Trans. Inf. & Syst.*, vol. E100.D, no. 8, pp. 1751–1761, 2017, doi: 10.1587/transinf.2016ICP0028.

[25] "All About BIND DNS: Who, How, & Why," *Linux Academy*, Jan. 14, 2020. https://linuxacademy.com/blog/linux/all-about-bind-dns-who-how-why/ (accessed Jul. 08, 2020).

[26] "DNSSEC Part II: the Implementation | Linux Journal." https://www.linuxjournal.com/content/dnssec-part-ii-implementation (accessed Jul. 08, 2020).

[27] "2. BIND Resource Requirements — BIND 9 documentation." https://bind9.readthedocs.io/en/latest/requirements.html (accessed Jul. 07, 2020).

[28] "20.4. The Configuration File /etc/named.conf." https://www.pks.mpg.de/~mueller/docs/suse10.1/suselinux-manual_en/manual/sec.dns.named.html (accessed Jul. 07, 2020).

[29] "14.6. DNS — Domain Name System." https://www.novell.com/documentation/suse91/suselinux-adminguide/html/ch14s06.html (accessed Jul. 08, 2020).

[30] "haveged(8) - Linux man page." https://linux.die.net/man/8/haveged (accessed Jul. 07, 2020).

[31] "DNS BIND9 Security Statements." https://www.zytrax.com/books/dns/ch7/security.html (accessed Jul. 07, 2020).

[32] S. Weiler <weiler@tislabs.com>, "DNSSEC Lookaside Validation (DLV)." https://tools.ietf.org/html/rfc5074 (accessed Jul. 07, 2020).

[33] "How To Set Up a Firewall Using FirewallD on CentOS 7," *DigitalOcean*. https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-firewalld-on-centos-7 (accessed Jul. 08, 2020).

[34] "DNS zone transfer," *Wikipedia*. Jun. 22, 2020, Accessed: Jul. 08, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=DNS_zone_transfer&oldid=963930305.

[35] O. Kolkman, "DNSSEC Key Management and Zone Signing," p. 26.

[36] "dnssec-signzone Command," Oct. 24, 2014. www.ibm.com/support/knowledgecenter/en/ssw_aix_71/d_commands/dnssec-signzone.html (accessed Jul. 08, 2020).

[37] "BIND DNSSEC Guide," p. 95.