DEPLOYING DNSSEC IN ISLANDS OF SECURITY

Submitted in partial fulfilment
of the requirements of the degree of

MASTER OF SCIENCE

of Rhodes University

Wesley Vengayi Murisa

Grahamstown, South Africa
March 2013

Abstract

The Domain Name System (DNS), a name resolution protocol is one of the vulnerable network protocols that has been subjected to many security attacks such as cache poisoning, denial of service and the 'Kaminsky' spoofing attack. When DNS was designed, security was not incorporated into its design. The DNS Security Extensions (DNSSEC) provides security to the name resolution process by using public key cryptosystems. Although DNSSEC has backward compatibility with unsecured zones, it only offers security to clients when communicating with security aware zones.

Widespread deployment of DNSSEC is therefore necessary to secure the name resolution process and provide security to the Internet. Only a few Top Level Domains (TLD's) have deployed DNSSEC, this inherently makes it difficult for their sub-domains to implement the security extensions to the DNS.

This study analyses mechanisms that can be used by domains in islands of security to deploy DNSSEC so that the name resolution process can be secured in two specific cases where either the TLD is not signed or the domain registrar is not able to support signed domains. The DNS client side mechanisms evaluated in this study include web browser plug-ins, local validating resolvers and domain look-aside validation. The results of the study show that web browser plug-ins cannot work on their own without local validating resolvers. The web browser validators, however, proved to be useful in indicating to the user whether a domain has been validated or not.

Local resolvers present a more secure option for Internet users who cannot trust the communication channel between their stub resolvers and remote name servers. However, they do not provide a way of showing the user whether a domain name has been correctly validated or not. Based on the results of the tests conducted, it is recommended that local validators be used with browser validators for visibility and improved security.

On the DNS server side, Domain Look-aside Validation (DLV) presents a viable alternative for organizations in islands of security like most countries in Africa where only two country code Top Level Domains (ccTLD) have deployed DNSSEC. This research recommends use of DLV by corporates to provide DNS security to both internal and external users accessing their web based services.

Acknowledgements

I acknowledge the financial support and time afforded to me to undertake my studies by the Zimbabwe Revenue Authority (ZIMRA) and their development partners AusAid. I also thank Professor Barry Irwin for his patience and guidance without which completion of this research project would have been very difficult. To fellow students, Walter, Mathe and other Security and Network Research Group (SNRG) students, I thank you for the invaluable support and assistance you provided to me. To my wife, daughter and family, thank you for the love, patience, support and prayers I have received from you during the entire duration of my studies.

Many thanks to all the friends at Rhodes and back in Zimbabwe who were there with kind words and encouragement. Finally, I thank the Lord for the strength, guidance and wisdom I received during my studies.

ACM Computing Classification System Classification

Thesis classification under ACM Computing Classification System (1998 version, valid through 2012)

C.2.2 [Network Protocols]: Applications (SMTP, FTP, etc.)

D.4.6 [Security and Protection]: Cryptographic Controls

E.3 [Data Encryption]: Public key cryptosystems

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computers and other devices on a network use Internet Protocol (IP) addresses for identification. It is however difficult for humans to use numerical IP addresses to identify hosts on the Internet as opposed to using host names that can be easily memorized. For this reason, the Domain Name System (DNS) (Lottor, 1987) was implemented as a way of making the Internet use easy for humans.

The DNS allows usage of human friendly names to identify computers and other devices on the network by humans. The names are often related to function and therefore easy to remember when interfacing with humans (Kurose and Ross, 2009). The DNS then converts the human friendly names to their numerical IP address for networks which can only use IP addresses to identify networked resources. A user, for example, keys in www.google.com to access the Google website and the DNS will convert the web address to its numerical IP address (e.g 74.125.233.17) and connect to the website.

At the time of designing the DNS much emphasis was placed on functionality at the expense of security (Kolkman, 2008). This is probably because at that time the Internet was a new technology and its use was not as widespread as it is today. Further, not many people possessed enough knowledge on computers, which meant that there were low levels of computer security threats.

Today, the Internet has turned the world into one global village where many people increasingly depend on it for everyday life (Perdisci *et al.*, 2009). In terms of security, the popularity of the Internet has also given rise to many security attacks on the Internet. Specific protocols like the DNS are exploited in these attacks. One reason that has made DNS vulnerable to security attacks is because DNS data is deemed public and cannot be hidden as it travels over the network, hence it can be easily intercepted and manipulated (Arends *et al.*, 2005a).

Major security flaws of the DNS were discovered in 1995 and were kept secret whilst, the Internet Engineering Task Force (IETF)[1] started researching on ways of providing security to the DNS (NLnetLabs, 2012a). The research resulted in publication of Request For Comments (RFC) 2065 (Kaufman and Eastlake, 1997) and subsequent RFC's (Arends *et al.*, 2005a)(Arends *et al.*, 2005b)(Arends *et al.*, 2005c) which defined a set of protocol extensions that provide integrity and authentication to the DNS. The Domain Name Security Extension (DNSSEC) introduced in March 2005, mitigates against many of the known threats against DNS. As reported by Atkins and Austein (2004), some of the main threats which include DNS query ID guessing and prediction, betrayal by trusted server, packet interception, denial of service, authenticated denial of domain names and name chaining are discussed in detail in Chapter 2.

RFC's are technical IETF publications detailing Internet specifications, research and standards (Daigle, 2010).

All of the above (together with other DNS related) threats are addressed by DNSSEC. DNSSEC provides integrity, authentication and non-repudiation by applying public key cryptographic systems to the DNS (Kolkman, 2009). In DNSSEC, data is signed with digital signatures to protect it against malicious modifications and provide non-repudiation services (Osterweil and Zhang, 2009). The entity receiving signed DNS data can use their public key to determine authenticity and integrity of the data it is receiving.

Despite DNSSEC being a solution to many threats encountered on the DNS, its implementation and adoption has been slow (Perdisci *et al.*, 2009). On of the reasons is that the root domain was only signed in 2010 and most of the TLD's remain unsigned (ICANN, 2012f). This means the DNS and subsequently the Internet will remain vulnerable to security attacks until all the TLDs and all the other sub-domains have been signed.

1.1 Problem Statement

The DNS is intended to provide accurate name resolution for Internet usage such as e-mail, web browsing, social networking, Internet banking and other e-commerce transactions among other uses. However, as already pointed out, DNS is vulnerable to attacks many of which can be prevented by widespread deployment of DNSSEC.

The deployment of DNSSEC by Top Level Domains (TLD's) is slow especially in Africa where only two Country Code Top Level Domain's (ccTLD's), Uganda's .ug and Namibia's .na are signed (ICANN, 2012e). This scenario makes it difficult for organizations in

[1] http://www.ietf.org/

countries where TLD's are not signed to deploy DNSSEC as they cannot publish their public keys through their parent domains. Such domains are said to be in islands of security. Inherently, this means users of the DNS continue to be exposed to various security attacks that capitalize on the weaknesses of the DNS. As stated in (Dagon *et al.*, 2009), absence of security on the DNS results in absence of security on the Internet and other network based services. This, in turn, given the increasing dependency on the Internet has dire consequences for many users and, indeed the world.

This project will investigate the methods of deploying DNSSEC especially in situations where an organization's TLD is not signed. The project will also analyze the tools currently available for implementing DNSSEC in an attempt to to come up with simplified DNSSEC deployment recommendations.

1.2 Research Objectives

This research will attempt to achieve four key objectives. It will :

1. Investigate the methods that can be used to deploy DNSSEC when the TLD is not signed.
2. Produce recommendations for deployment of DNSSEC for Internet and intranet users whose TLD is not signed.
3. Attempt to find some of the reasons behind the slow uptake of DNSSEC.
4. Find out the effect of DNSSEC on DNS message size.

1.3 Research Scope

The two widely used computing platforms are Windows and Linux (Whaley *et al.*, 2010). For this reason, a decision was made to limit the scope of this research to name server applications used on these two platforms. Being the most stable and widely used implementations of the DNS, Windows Server 2008 (Microsoft, 2008) and Bind version 9.8 (ISC, 2012) were used for this research.

In terms of focusing the investigator to the research objectives, the following questions were used:

1.4 Research Questions

The following questions will be used to guide the research work towards attainment of its objectives.

1. What does it take to deploy DNSSEC in islands of security?
2. What are the benefits of DNSSEC to an average user?
3. What are the DNSSEC tools and applications available?
4. What is the impact of DNSSEC on DNS traffic and message size?
5. What are the costs involved in implementing DNSSEC?

These questions were regarded as critical to the attainment of the research objectives. As such addressing them effectively provides the scope for this research.

1.5 Document Structure

The remainder of the thesis is arranged as follows:

Chapter Two gives a brief history of the Internet and networking as well as the DNS. It also explains the components that make up the DNS before providing an analysis of historical and current security threats to the DNS.

Chapter Three outlines the new components brought by DNSSEC to the DNS. It also presents the benefits, deployment statistics of DNSSEC and the challenges encountered in its deployment. The chapter also looks at the expected future uses of DNSSEC.

Chapter Four explores the practical implementation of name servers used in the research and the tests conducted are elaborated.

Chapter Five presents and evaluates the results of the tests done. Further, the chapter provides a brief perspective on how these results can be used to promote DNSSEC.

Chapter Six provides concluding remarks for the research. It reflects on how the results answer the research questions.

Chapter 2

The Domain Name System

The Domain Name System (DNS) provides name resolution for Internet usage. To understand both how DNS works and some of its vulnerabilities, this chapter will, among other things look at the historical evolution of the DNS. Further the chapter will look into the key components and security threats of the DNS.

2.1 Motivation for DNS

The Internet, (then referred to as the ARPANET) was created by the US Department of Defense in 1968 to research on computer networking and its major objective as mentioned by Roberts (1986) was to establish a network that could allow sharing of computer programs and databases, provide load balancing services and electronic mail systems as well as remote accessibility. Another objective of the agency was to share other expensive computer resources and allow collaboration by important research institutions in the US (Liu and Albitz, 1998).

The ARPANET expanded in size as its usage continued to increase and it increasingly became difficult for humans to use IP addresses for the many machines on the network. This resulted in the development of a naming scheme where computers and other devices were given names that could be easily memorized by people. Humans would use these names and the networking system would convert these human names to their respective IP addresses. The names were easier to memorize as they reflected functionality or other property of the resource and were changeable.

This system also provided system administrators with a way of making technical changes to devices on the network without making the end user aware of these changes (Aitchison,

2005) . This means, IP address changes could be made to shared devices like file servers and printers without the users knowing it because they will be using the host name which remains static.

2.1.1 Hosts File

The list of host names and their respective IP addresses were originally stored in a text file called the 'hosts.txt' which was distributed to all computers on the network for use in name resolution (Liu and Albitz, 1998). Printed copies of the file would then be distributed to systems administrators (Neigus and Feinler, 1973). The description of the file, and the instructions and credentials for obtaining the file together with the naming and addressing convention for network devices is documented in RFC 952 (Harrenstien *et al.*, 1985). Whenever a user referred to a device on the network with the host name, the computer would locate the host name on the hosts.txt file and find the device's corresponding IP address. The computer would then use that IP address for the communication process with that host.

2.1.2 Problems Associated with Hosts File

The manual copy of the hosts.txt file resulted in inconsistencies on the network due to the rapid changes of computers on the network. As a result, different networks usually had different copies of the host name file. This led to the development of an online copy of the hosts.txt documented in RFC 606 (Deutsch, 1973) and RFC 608 (Kudlick, 1974). The online hosts.txt file was an ASCII file with host names, address and other attributes. It was maintained online and updated periodically or when necessary. Users obtained the file from the FTP server specified in the document. .

The growth in the size of the ARPANET resulted in proportional growth in the size of the hosts.txt file. This resulted in a large hosts.txt file that used up a significant amount of network bandwidth as all computers on the ARPANET tried to get an updated version of the file. This also used a lot of processing power of the computer hosting the text file whilst managing many FTP sessions. This amount of bandwidth, quantified by RFC 1034 (Mockapetris, 1987a) as "the square of number of hosts in the network" and processing power consumed by this process made the system expensive to maintain.

It increasingly became difficult to maintain consistency of centralized hosts name system because at any given time host updates were being received and retrieval operations by other hosts were taking place (Aitchison, 2005). Retrieving an address for a single host from a list with hundreds of thousands of address became very slow. It also became difficult to maintain unique names on the hosts.txt file as there were many people adding new computers at the same time and name collision became another serious problem with the centrally managed database of names (Liu and Albitz, 1998).

System administrators were responsible for maintaining their own LAN and there was no way of preventing identical names in different LANs hence identical names could be submitted by different organizations. This led administrators of the ARPANET to decide that the distributed nature of the Internet required a distributed host naming system and research into alternative naming system began.

2.2 Evolution of the DNS

The NIC maintained hosts.txt file was useful for providing a computer naming system in the ARPANET. The system started giving problems when the network expanded in size as the ARPANET evolved into the Internet. This resulted in the need for developing a decentralized, multi-purpose naming system that could sustain the growing network. The initial research work in a new name server system is documented in several Request For Comments (RFC's) that include RFC 819 (Su and Postel, 1982) and RFC 830 (Su, 1982).

These documents presented different proposals for the name server system but they had a common design proposal of a tree structured global name system that represents the structure of the organizations making up the name server system. Thus the need for a distributed name server system became a design objective of the DNS. The other objectives were consistency, independence from network design, to be useful for many different purposes and types of computers and devices. All of them also proposed the use of a dot (.) to separate the different levels of the hierarchical name structure. The work evolved into RFC's 1033 (Lottor, 1987), 1034 (Mockapetris, 1987a) and 1035 (Mockapetris, 1987b) which define the basis of DNS as we know it today.

The next subsections will elaborate on the finer details of the DNS as specified by the RFC's 1033, 1034 and 1035.

2.2.1 DNS Authority and Delegation

Critical to the DNS system is the concept of authority and delegation (Aitchison, 2005). For every node that is on the hierarchical structure of the DNS, there is an organization that has authority for managing and operating that particular part of the domain name system. The authority for any node can, in turn, delegate authority to lower levels(Liu and Albitz, 1998). At the top of the domain name system tree is the root domain which is the highest level in the hierarchy.

The root domain is operated by Internet Corporation for Assigned Names and Numbers (ICANN), which retains authority for the whole DNS (ICANN, 2012b). The root domain was previously operated and managed by the US Department of Commerce, but it was later handed over to ICANN under the Cooperative Research and Development Agreement (CRADA) in 1998. ICANN entered into an agreement with other organizations for the maintenance of the 13 logical and geographically spread root servers that make up the root domain (IANA, 2012b).

The root domain delegates authority to two top level domains (TLD's), that is generic TLD (ggTLD) and country code TLD (ccTLD) (ICANN, 2012b). Below these two TLD's are several other delegations to other secondary and tertiary level domains which vary on a country by country basis as shown by the diagram below.

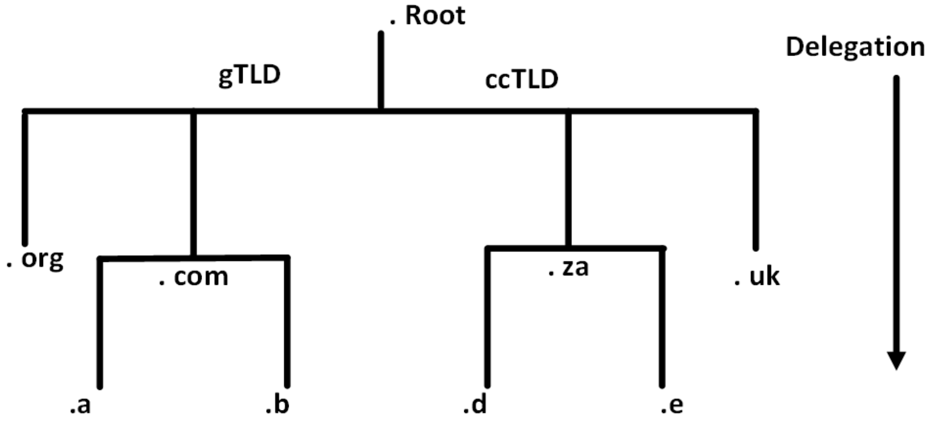


Figure 2.1: Domain Structure and Delegation

2.2.2 Components of the DNS system

The domain name system is made up of three components that describe the initial design of the DNS. A description of these three which are domain name space and resource records, name server and resolver is outlined below:

1. **Domain Name Space and Resource Records** - The domain name space is the entire collection of all the domains arranged in a hierarchical structure (Bau and Mitchell, 2010). The hierarchical structure has many nodes with each node

describing the global properties, hosts and services provided by that part of the name server system. The collection of all the leaves and nodes in this tree structure make up the global domain space. The definition of characteristics and properties of node in the domain space is achieved by standard textual descriptions of host and services called resource records (RR) (Bau and Mitchell, 2010). The RR are contained in standard text files called zone files. Zone files are text files editable by any standard text editor and contain three type of entries namely comments, directives and resource records. Resource records are not required to be in any order in a zone file.

2. **Name Server** - Is a software program that provides storage and manages data that make a domain (Lioy *et al.*, 2000). The domain data is divided into zones and each name server should have at least one zone. The name server can have many optional functions but its primary function is to provide query resolution using data in its zones (ISC, 2012). The name server can also be configured to provide other optional functions like recursive query resolution which will be explained later in this chapter. It is also a requirement that there should be at least one redundant name server for each domain implementation (Aitchison, 2005).

   This is a way of providing backup in case of failure of the primary name server and many organizations in practice implement more than one name server. It is also recommended to have name servers available in different geographical areas to further improve availability and reduce network bandwidth usage. Name servers can be classified into different groups depending on functionality and how they are implemented (ISC, 2012). Examples of different types of name servers include master name server, slave or secondary name servers, forwarding name servers, stealth name server and authoritative only name server.

3. **Resolver** - Is a software program that is designed to make inquiries about parts of the domain space from name servers in response to client requests (Lioy *et al.*, 2000). The resolver need to have access to at least one name server for it to be able to process user queries. The resolver will get answers from the name server it has access to or it will use that name server to pursue the query with other name servers through referrals when it does not have that information itself. It is often implemented to resolve names on behalf of client computers (Chandramouli and Rose, 2010).

   Client computers implement a “stub resolver” that sends DNS queries to resolvers for query resolution since the resolver is a complex program that is not commonly implemented in client operating systems like Windows and Unix (van Rijswijk-Deij, 2012). A resolver is different from an authoritative name server in that it does not have any zone files but it uses authoritative name servers to provide answers to clients. Some DNS systems combine an authoritative name server and resolver in one implementation.

2.3 DNS In-depth

This Section describes the underlying details of the DNS protocol which include how domain names are formulated, how information is stored in resource records and how messages are transported from one name server to the other.

2.3.1 Domain Names

Domain names are made up of labels of nodes in the domain tree read from left to right and separated by dots (.) (Bau and Mitchell, 2010). That is, a domain name is obtained from the domain name tree by starting to read the least significant node label, separate it from the next node by placing a dot and then read the name of the next label until you reach the most significant node which is the root domain.

The rules used to construct domain names in the ARPANET were adopted in the DNS naming conventions for backward compatibility (Lottor, 1987). These naming rules state that a domain name should start with an alphabetical letter and end with either an alphabetical letter or a number. The characters in between the name can be alphabetical letters, numbers or the hyphen.

There are also a size limitation in the domain names with each label size restricted to 63 characters or less while the full domain name cannot be more than 255 characters (Lottor, 1987). The characters used in the domain tree are not case sensitive such that a given name is treated as one name if its in upper case or lower case (Liu and Albitz, 1998). The rules also specify that the mail exchange name will replace the @ symbol with a dot. For instance, a mail exchange address administrator@example.com will translate to administrator.example.com.

2.3.2 Resource Records

As explained previously in this chapter, resource records (RR) contain information about resources and services within a certain domain and are stored in any order in a zone file

(Yang *et al.*, 2011). They can be listed in a single line terminated by semi-colon or they can take up several lines when parenthesis's are used (Mockapetris, 1987a). The RR has five properties which are listed below:

1. **Owner** is the domain name to which the RR belongs to and is usually implicitly defined once in the Start Of Authority (SOA) record. When the owner value is not specified, the owner defaults to the value for the previous record.
2. **Type** refers to predefined resources record types codes. The commonly used RR types codes defined in RFC 1034 and 1035 are listed and described in the table 2.1 (Ariyapperuma and Mitchell, 2007).
3. **TTL** is an integer value representing time in seconds a RR is cached by resolvers before it is discarded. TTL can be used to minimize or prevent caching altogether and a zero value for TTL like is the case with most SOA resource records means that the RR is never cached (Son and Shmatikov, 2010).
4. **Rdata** contains additional data that describes the RR (Mockapetris, 1987a). The value for Rdata and can be dependent on the RR type and class. For example where a RR is of type A and class IN, the Rdata Section will be a 32 bit IP address for the resource.

Table 2.1: DNS Resource Records

Resource Record	Description
A	Address. Defines an IPV4 IP address of a host in the domain.
CNAME	Canonical name. An alias for a host in the domain.
MX	Mail Exchange. Specifies the domain's mail exchange.
NS	Name Server. Identifies the authoritative name servers for a domain.
PTR	Pointer. Holds a host IP address and is used for reverse mapping.

2.3.3 Message Transportation

The transportation of DNS messages can use datagrams or virtual circuits. Datagrams are normally preferred for queries because of their low overheard and better performance

(Kurose and Ross, 2009). Virtual circuits can be used for any DNS activity including zone transfers because they are more reliable than datagrams.

Two protocols identified for DNS message transportation are UDP and TCP (Mockapetris, 1987b). Port 53 is specified for both transmission protocols. UDP protocol is recommended for usage in standard DNS queries because it has a better turnaround time as compared to TCP (Kurose and Ross, 2009).

However, UDP has a 512 byte size limitation where as TCP does not have that size limitation (Liu and Albitz, 1998). DNS messages using the UDP are truncated if they exceed this limit and a TC bit in the message will confirm the truncation. Usage of TCP in DNS message is useful for other DNS operations which involves messages larger than 512 bytes like zone transfers. It is however important to configure networking equipment like routers and firewalls to allow bigger message sizes as some of them are preconfigured to limit the size of DNS messages as detailed in a CISCO report on DNSSEC (Eklov and Lagerholm, 2010). This may result in implications for DNSSEC where messages can be as big as 4096 bytes.

2.3.4 DNS Queries

DNS system queries can be classified onto three groups namely recursive, iterative and inverse queries as detailed below (Liu and Albitz, 1998):

1. **Recursive Queries** perform all the tasks that are required to find a complete answer to a query. A name server that implements recursive behavior will first try to answer the query from its local records. If it fails to get the answer locally it will follow referrals to other name servers from the root servers until it gets a final answer to the query. If it cannot provide the answer due to some problems, an appropriate error message is returned; typically a non-existence or temporary unavailability error message (Alexiou *et al.*, 2010).

   DNS standards do not require name servers to be recursive, but recursion is an important feature that improves performance of the DNS system by making available previously used RR subject to their TTL (Yang *et al.*, 2011). Recursive behavior in a name server is useful in instances when the requester (e.g. stub resolver implemented by client operating systems) can only use a direct answer to its query. When recursion is required, it is denoted by a RD bit in the message header. A server that has caching capabilities sets the RA bit in the message header (Mockapetris, 1987b). Recursive name server are used to handle all queries in a domain thereby protecting the authoritative name server from the security risks associated with answering queries and caching responses.

2. **Iterative Queries** are different from recursive queries in that name servers implementing iterative behavior do not follow up a query until they get a final answer, instead they use data in their local zone files and caches to answer queries (Aitchison, 2005). If the requested information is present in these two sources, the answers are provided, otherwise non-existence or temporary unavailability errors are sent as in recursive queries. If the server does not have information in its local sources it will not look for the information from other name servers like in recursive queries but answers with a referral to other servers that may have the answer or are nearer to the answer. Iterative queries help reduce performance overhead for busy name server which take a huge number of queries like the root server. Iterative queries prevent caching of RR,and this prevents security risks that are associated with caching behavior like cache poisoning which will be explained later in this chapter.

3. **Reverse queries** are specified in RFC 1035 (Mockapetris, 1987b) but are obsoleted by RFC 3425 (Lawrence, 2002). The implication of this RFC is that the IQUERY opcode 1 defined in the message format header Section becomes obsolete as well and this RFC recommends that IANA should permanently retire opcode 1 and not assign it to any use. This means that any name server that receives a query with an opcode 1 will respond with a not implemented error message.

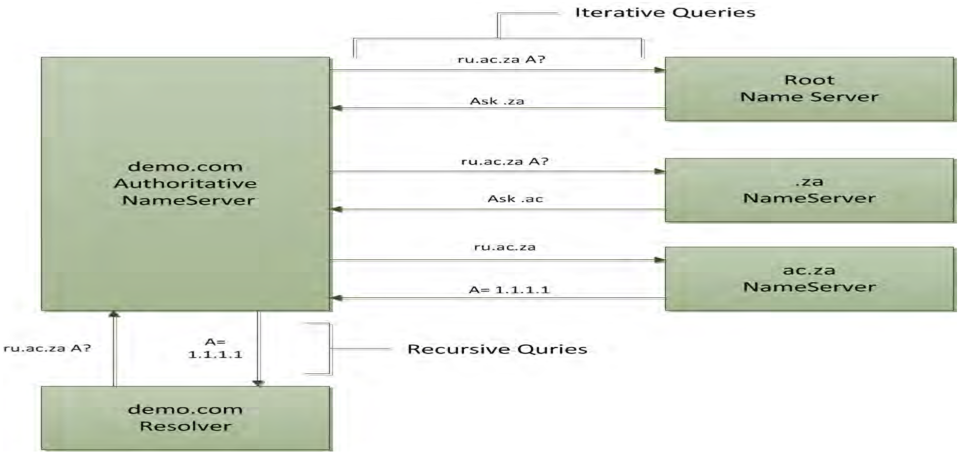


Figure 2.2: Iterative and Recursive Queries

2.4 Security Weaknesses of the DNS

The DNS is vulnerable to many security threats mainly because its initial design did not incorporate any security mechanisms (Son and Shmatikov, 2010). In addition, UDP which is used for DNS message transportation does not have adequate security measures for the message it is transporting (Kurose and Ross, 2009). Further, the UDP does not have mechanisms to guarantee message authenticity and integrity.

If a request or response is lost or delayed along the communication channel, the DNS issues another request and is therefore not able to handle packet delays (Tzur-David *et al.*, 2009). The DNS also accepts and caches the first response it gets for a query it has issued (Santcroos and Kolkman, 2007). This characteristic, though noble for fast transportation of messages, makes the DNS vulnerable to various security threats.

Before analyzing the security risks associated with the DNS, it is important to understand the potential sources of the security threats. In Figure 2.3 every numbered flow of data represents a potential security threat. Number 1 depicts zone files that can be corrupted maliciously or accidentally. Unauthorized dynamic updates and zone transfers can occur between primary and master server as shown by numbers 2 and 3.

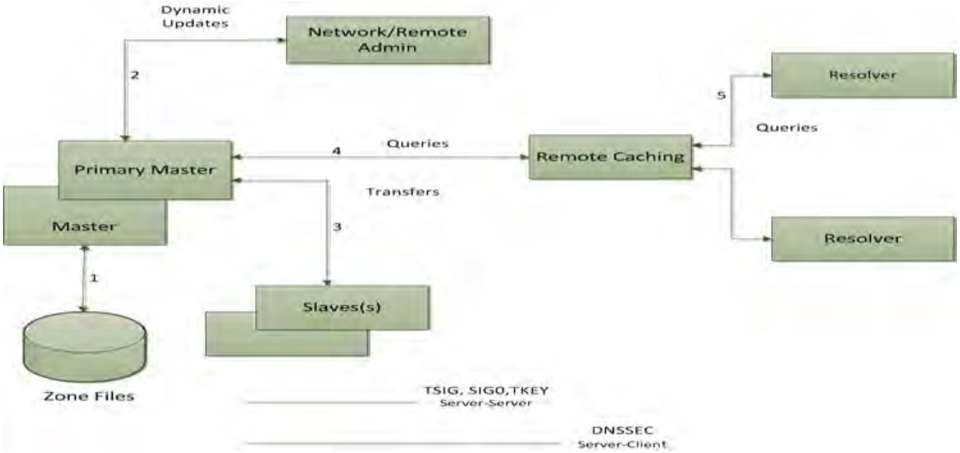


Figure 2.3: DNS Security Overview after Aitchison (2005)

Number 4 represents possible cache poisoning attacks against recursive name servers through several attack methods, such as IP address spoofing and data interception. Queries or responses from client stub resolvers to caching resolvers labeled number 5 on the diagram above can also be intercepted and spoofed in the communication channel.

An elaboration of the common known threats is provided in the next subsections.

2.4.1 Packet Interception

It refers to the interception of DNS messages between two hosts by an attacker in the same network as the victim host or a compromised resolver, as illustrated in Figure 2.4 (Atkins and Austein, 2004). Packet interception maliciously obtains message ID, source IP address, port number and question. It uses these parameters to send a forged response to a DNS client which accepts the first response with the correct parameters (Davies, 2008).

An attacker who successfully intercepts DNS messages compromise the confidentiality of the message by eavesdropping (Green, 2005). Message integrity is also compromised if the attacker modifies the message. If a message is destroyed or causes one of the hosts to end communication availability is compromised. These attacks are mainly facilitated by the widespread usage of wireless networks which can be easily hacked (Santcroos and Kolkman, 2007).

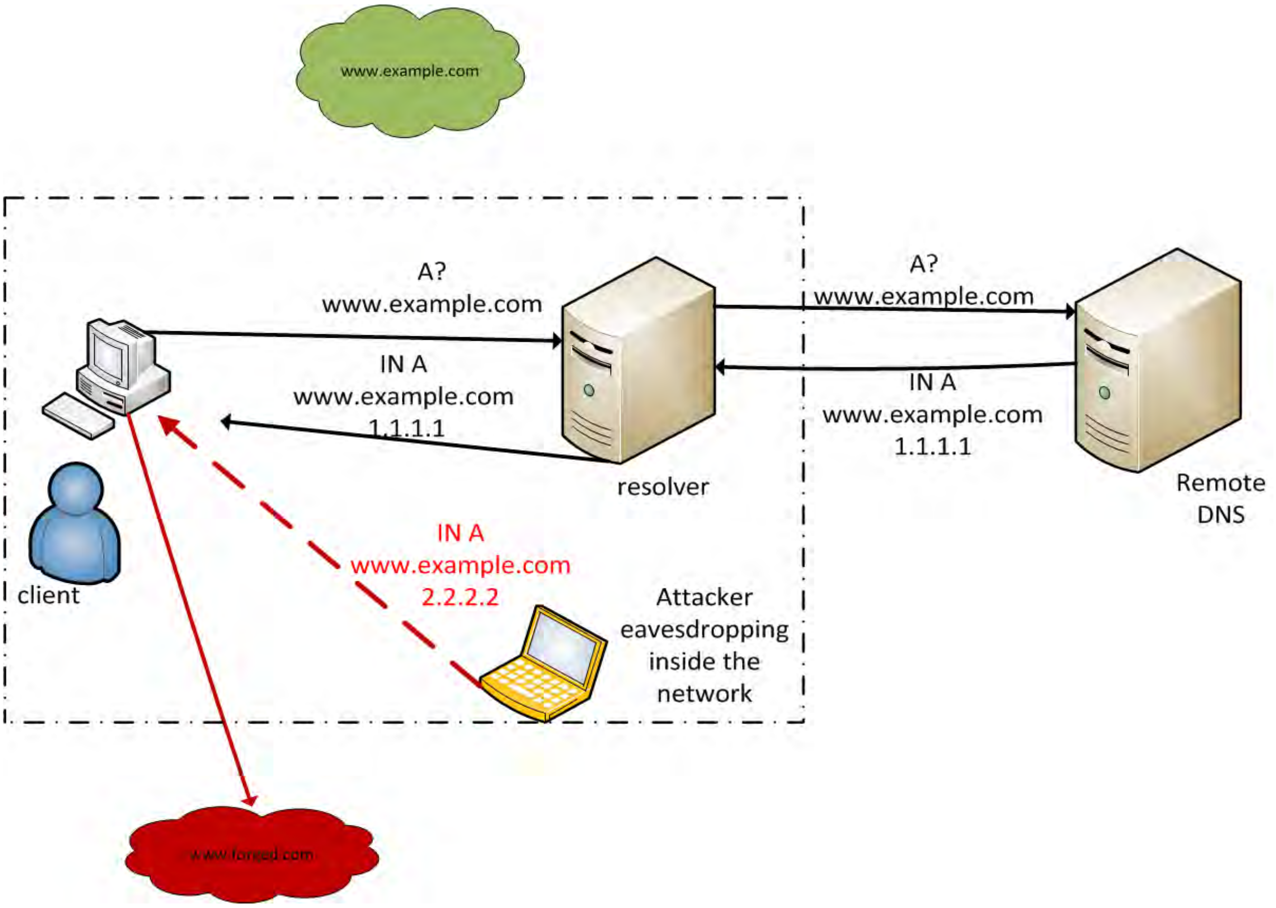


Figure 2.4: Packet Interception

Man-in-the-middle (MITM) and eavesdropping combined with spoofing attacks are all different forms of packet interception attack and they depend on sending a response to the resolver faster than the legitimate response (Atkins and Austein, 2004). In the example illustrated in Figure 2.4, the attacker first intercepts the client query and sends a spoofed response to the client which is seeking IP address of www.example.com earlier than the resolver. The client will use the spoofed address believing it to be true.

DNS spoofing involves the insertion of incorrect name resolution data to legitimate responses by a malicious host as illustrated in Figure 2.4 (Babu *et al.*, 2010). It can be used to obtain information that can be used for other security attacks such as phishing (Tzur-David *et al.*, 2009). For further elaboration of this attack, assume, a telnet client asks its resolver for the IP address of telnet server A. The DNS message is intercepted and it receives a spoofed answer. When it initiates a TCP connection to the spoofed telnet server address, the user's login credentials are stolen and the connection is dropped.

2.4.2 ID Guessing and Query Prediction

The message ID and message port are 16 bit fields each, resulting in a possible 2^{32} combinations which can be obtained through brute force attacks (Granstrom, 2009). Furthermore, the message ID and port number can also be obtained by observing previous traffic. The port number is usually fixed due to firewall restrictions and also because most name server applications use a fixed port number (Liu and Albitz, 1998). This effectively reduces the possible combinations to 2^{16} thereby making the brute force attack more easier. The possible combinations can even be lower if the name server does not use the entire field for message ID (Santcroos and Kolkman, 2007).

The query name (QNAME) and query type (QTYPE) can be obtained by an attacker who invokes a name lookup of their choice. This is accomplished by sending a query of the attackers choice to the target resolver (Andersson and Montag, 2008). The attacker may also be aware of a query from a third party hence it becomes aware of the QNAME and QTYPE (Lioy *et al.*, 2000). With the port number and message ID obtained as previously explained the attacker can then inject a bogus response to a resolver. The DNS will accept the spoofed response, cache it and give it out to other clients requesting the same domain address.

This attack is similar to packet interception in that both attacks are done by providing bogus response to queries being transmitted over the network. However, packet interception is done by an attacker in the same network as the victim resolver whereas id guessing and query prediction attacks are done outside the network of the target resolver as illustrated in Figure 2.5 (Atkins and Austein, 2004). The attack however works only

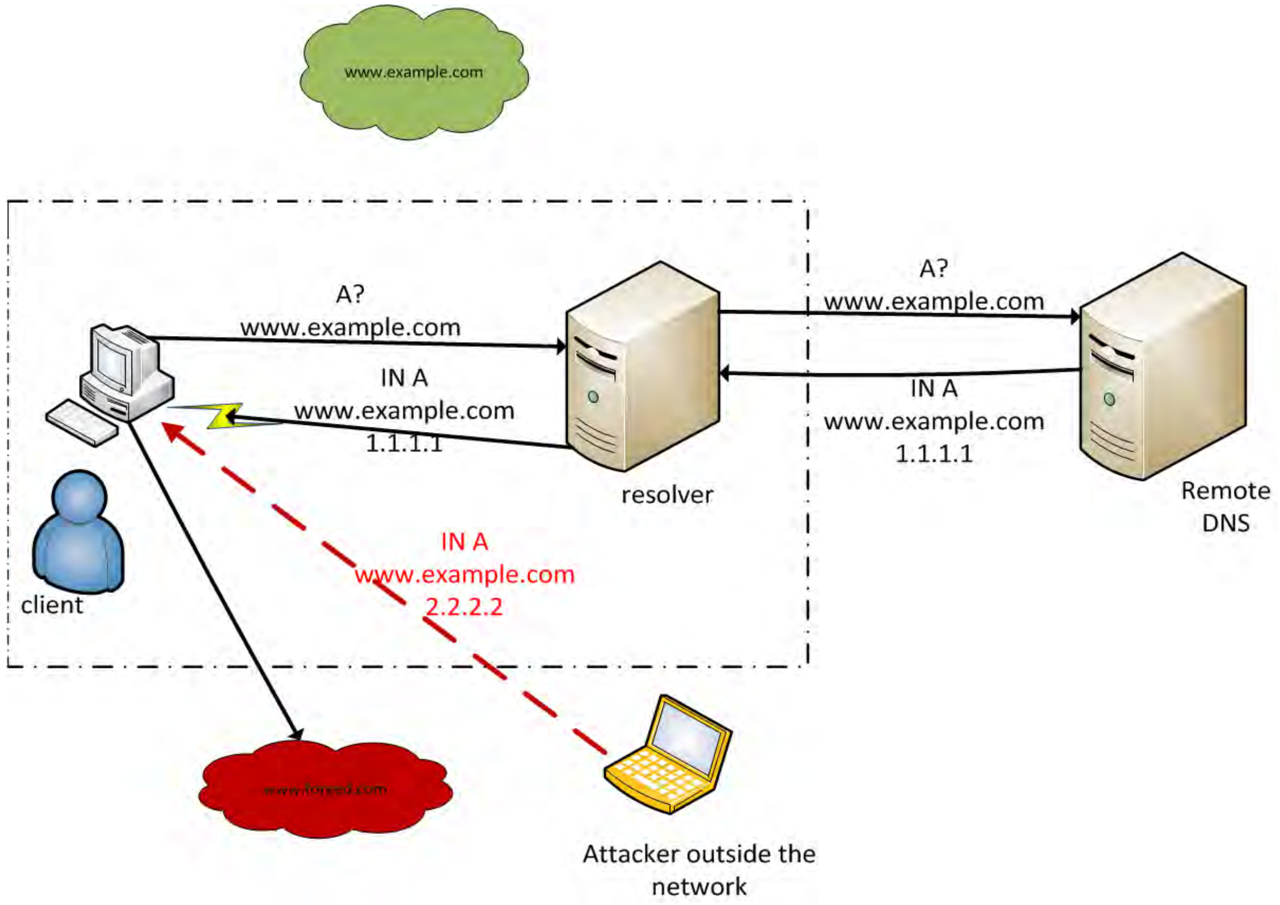


Figure 2.5: ID Guessing and Query Prediction

when the attacker makes a correct guess or prediction of the message parameters of the target resolver.

2.4.3 Cache Poisoning

Cache poisoning involves inserting forged DNS query responses to a recursive name server's cache using various attack methods that include the two attacks elaborated in Sections 2.4.1 and 2.4.2 above. Even though different methods can be used to force a name server to resolve a targeted domain name to a malicious IP address, it is the insertion of forged addresses in a caching name server that is referred to as cache poisoning.

Stub resolvers depend on recursive DNS (RDNS) resolvers for mapping domain name to IP addresses. The RDNS performs a lookup of the domain name from other authoritative name servers, provides the answer to the stub resolver and keeps the response for a time period defined by the resource record's TTL (Perdisci *et al.*, 2009). A successful cache poisoning attack will provide a forged response that arrives at the resolver earlier than the legitimate response, since recursive servers accept any answer which arrives first as shown in Figure 2.6.

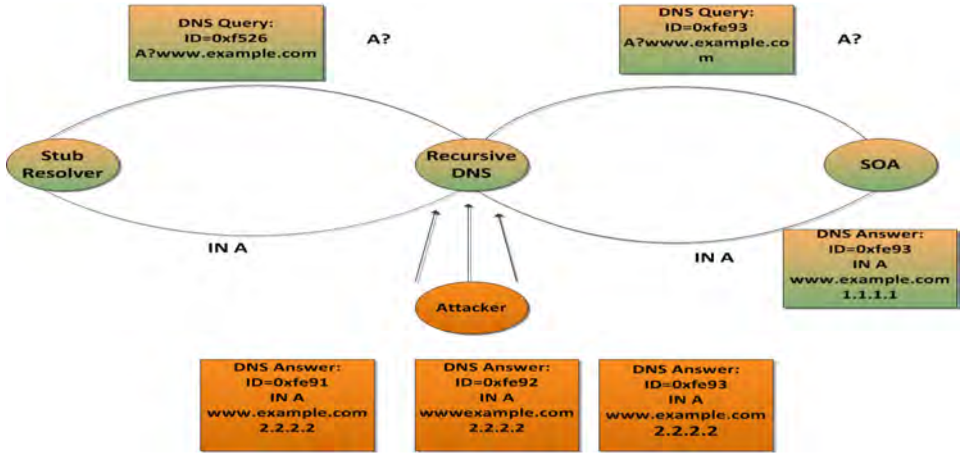


Figure 2.6: DNS Cache Poisoning (Dagon *et al.*, 2008)

In Figure 2.6, the stub resolver asks the resolver for the IP address of www.example.com but before a legitimate response is obtained from the legitimate authoritative name server, a remote attacker provides a series of forged response with different message ID's. The resolver will accept the response with the correct message id (0xfe93 in this example), provided the the attacker gives the correct message parameters because the resolver accepts the first message with the correct parameters. The forged response will then be cached in the resolver and subsequently sent to DNS clients. The legitimate responses received after the forged one has been accepted will be dropped.

Cache poisoning can be a gateway to further attacks like pharming. In pharming, the malicious address can be used to redirect traffic to a bogus website for identity theft, distribution of malware, dissemination of false information and man-in-the-middle attacks (Tzur-David *et al.*, 2009).

2.4.4 Kaminsky Cache Poisoning Attack

It is a variation of the normal cache poisoning attack that uses brute force techniques to pollute the cache of a name server with malicious IP address for NS resource records (Perdisci *et al.*, 2009). The attack works by making an unlimited number of queries for non-existent domain names to increase chances of making a correct transaction ID guess (Kaminsky, 2008). The 'Kaminsky' attack does not typically target the answer field of a query. Rather, it aims at providing a malicious address for the authoritative name server through the Rdata Section of a query response (Dagon *et al.*, 2009).

Figure 2.7 illustrates this attack on the domain example.com. The attacker makes a query to the targeted resolver for www.example.com. A large number of responses that attempt to make a correct guess of the message ID are sent to the targeted resolver. The attack continues to make queries for non-existent labels such as 1.example.com, 2.example.com until a correct guess is made. The greater the number of these queries, the greater the chances of correctly guessing the message ID. When a correct guess is made through NXDOMAIN response, the Rdata Section of the response will contain forged example.com NS IP address and this is cached by the resolver. The forged IP address and will be used to redirect queries for the example.com domain to the forged name server (Son and Shmatikov, 2010).

Kaminsky (2008) illustrated that if an attacker sends 100 spoofed responses to a DNS query, the probability of success is reduced from 1/65000 to 1/650 chances of success. DNS software that is able to generate and use random port numbers instead of the common port 53 for their messages can reduce the expected time of a successful attack from seconds to tens of minutes (Bau and Mitchell, 2010)

2.4.5 Birthday Attacks

Birthday attacks are another variation of cache poisoning attack. They take advantage of the low entropy of the message ID to mount brute force attacks against recursive name servers. They work by convincing recursive name servers to make multiple and concurrent queries for a single domain name and for each query generated, then provide multiple spoofed responses for each query generated (Dagon *et al.*, 2009). The increased number of responses to the queries greatly increases chances that one of the spoofed response's transaction ID will be a correct guess.

The birthday attack obtains a high level of success from relatively small number of packets. For example, 300 packets guarantee 50% success, 750 packets guarantee a 99% success rate whilst a regular spoofing attack has a 1.14% success rate with 750 packets. Suppressing recursive name servers from making multiple queries for a single domain and use of pseudo random generated transaction ID has been proposed as a solution. Bind 9.5 has implemented this solution but Tzur-David *et al.* (2009) argue that all Bind versions are still vulnerable to this attack.

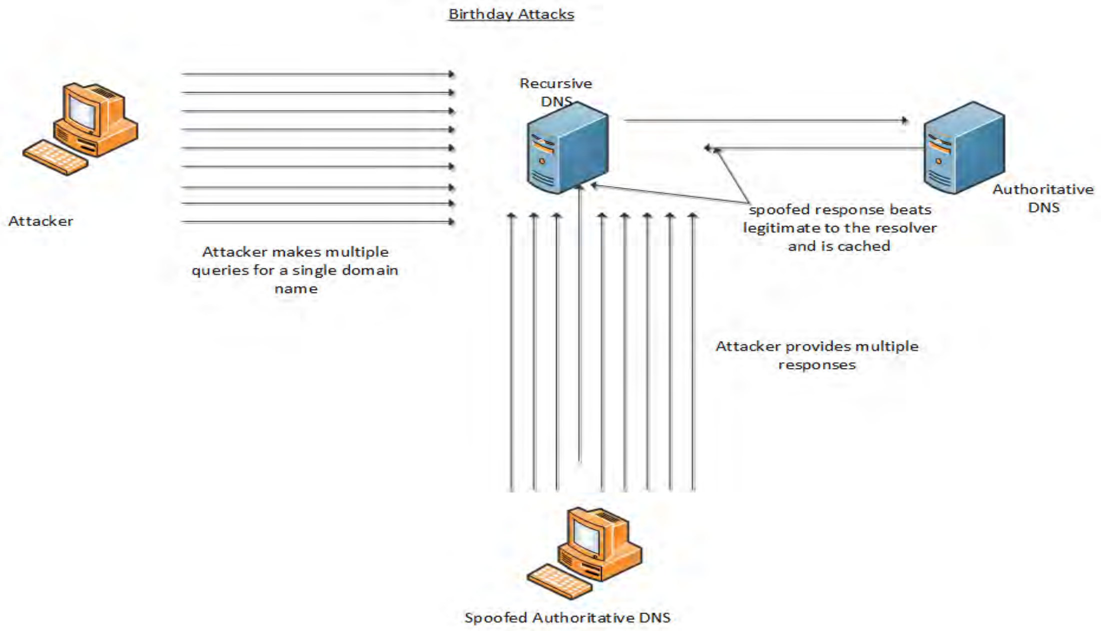

Birthday Attacks
Attacker
Attacker makes multiple queries for a single domain name
Recursive DNS
Authoritative DNS
spoofed response beats legitimate to the resolver and is cached
Attacker provides multiple responses
Spoofed Authoritative DNS

Figure 2.7: DNS Birthday Attacks (Ollmann, 2007)

2.4.6 Denial of Service (DoS) and Distributed Denial of Service (DDoS)

Any network service is vulnerable to denial of service (DoS) attack and the DNS is not an exception in this regard (Atkins and Austein, 2004). For DoS attacks to be effected, recursive name servers are flooded with malicious queries and as a result fail to service legitimate name lookups. DNS is prone to these kind of attacks because they accept source-spoofed packets and use open resolvers (DNS-OARC, 2012b). Attackers also take advantage of known software bugs in specific name server software and send malformed messages that will cause a name server to crush (Kristoff and Joffee, 2007). DoS attacks are usually launched from a single host and are targeted at one name server (Santcroos and Kolkman, 2007).

The reason for launching DoS attacks is to cause disruptions in the name resolution process and internet services (Kristoff and Joffee, 2007). An attacker targeting an application available over the internet can cause the name resolution process to fail using DoS attacks instead of directly attacking the application. DoS attacks can be part of a larger attack involving extortion, hijacking or physical damage. For example, in August 2012, a large mobile services provider in the US called AT & T was subjected to denial of service attacks that disrupted customer service for several hours (Williams, 2012).

Distributed denial of service attacks (DDoS) occur when DoS attacks are launched from

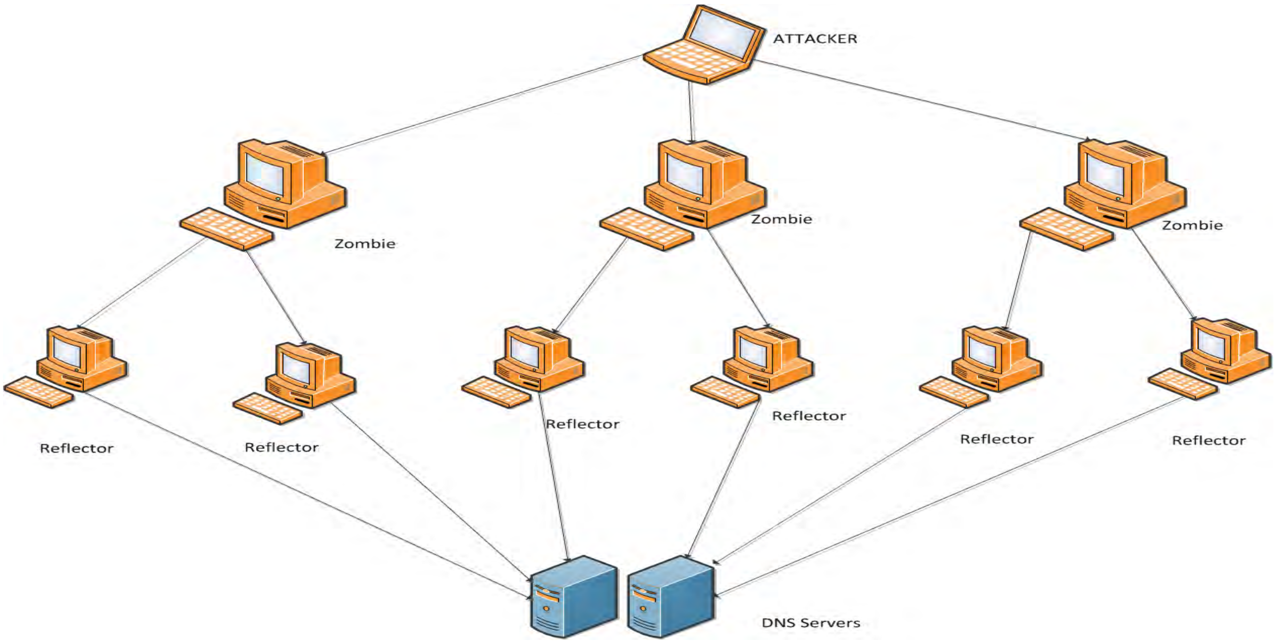


Figure 2.8: DDoS Attacks

more than one host. Figure 2.8 shows an attackers using a network of remotely controlled robot computers called bots to launch large scale DDoS attacks. Botnets are a network of remotely controlled computers compromised in many different means such as e-mail borne worms that install applications used to remotely control the computers (ICANN, 2006). The zombie computers that make up the botnets can be used to launch a packet flooding DDoS attack against a target DNS server (Kristoff and Joffee, 2007).

2.5 Summary

This chapter provided a brief historical perspective that reflects how the DNS was developed. It also described the components that make up a zone and the operations involved in managing a domain. An overview of some of the security threats affecting the DNS was also provided. The next chapter will explain how these threats can be mitigated by security extensions to the DNS.

Chapter 3

Domain Name Security Extension

This chapter describes the structure of DNS Security Extensions and how it mitigates against the DNS security threats outlined in Section 2.4. The deployment statistics of DNSSEC and possible reasons for slow uptake are presented in Section 3.4. In addition, this chapter outlines arguments for deploying DNSSEC before analyzing the future uses of DNSSEC in Sections 3.6 and 3.7. The chapter closes with an analysis of how DNSSEC mitigates against the DNS security threats outlined in Section 2.4.

3.1 DNSSEC Overview

The previous chapter on DNS ended with an analysis of DNS security threats that exposes the name resolution process and the Internet to security attacks. Cryptographic system based solutions such as Domain Name Security Extensions (DNSSEC) provide long term solutions to the security of the Internet and other network services that depend on the DNS for name resolution (Dagon *et al.*, 2009). DNSSEC, as the name suggests, was developed to include security features that were lacking in the original design of the DNS. Leading US security researcher Kaminsky (2008) specifically called for deployment of DNSSEC to protect against the cache poisoning attacks he discovered.

DNSSEC uses asymmetric keys, namely: the Zone Signing Key (ZSK) and the Key Signing Key (KSK) to encrypt the data in a zone as shown in Figure 3.1. The ZSK signs all the resource records before its public key portion is signed by the KSK and stored in a new RR called the DNSKEY (Bau and Mitchell, 2010). The public key of the KSK is stored in another new RR called DS and is sent to the parent domain for use in verifying the zone's resource records. Whenever the keys are changed, the child should provide the updated DS records to the parent, otherwise validation fails. A security aware name server responds to a query from a signed resolver with the requested data and the corresponding resource record signature (RRSIG) (Osterweil *et al.*, 2008). The resolver then uses DNSKEY of the zone to verify the authenticity and integrity of the response received.

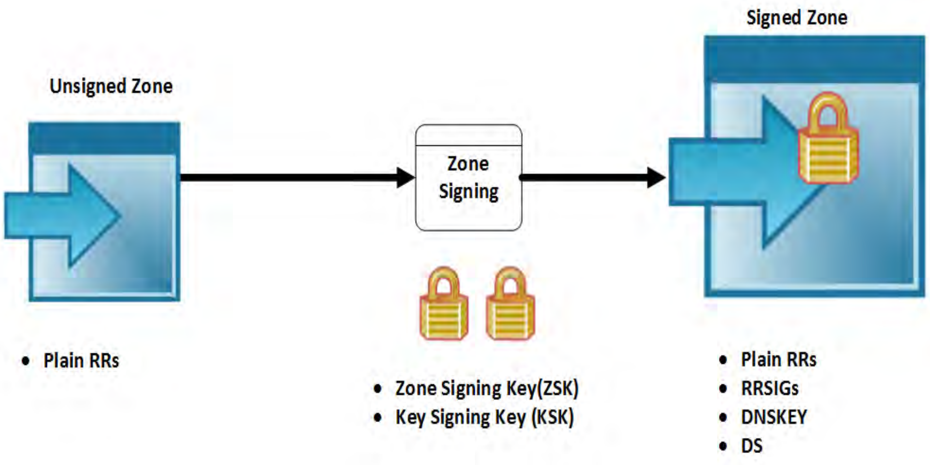


Figure 3.1: DNSSEC Illustration

Only security-aware name servers can understand DNSSEC and verify the integrity and authenticity of query results. However, DNSSEC provides for backward compatibility with unsigned name servers which get responses without security data. The unsigned data is typically subject to the DNS security risks mentioned in Chapter 2 hence the need for all name servers for a domain to be signed.

DNSSEC uses a chain of trust in which every parent domain except the root authenticates child zone data using the DS resource record (Gieben, 2004). A child domain whose parent is not signed is said to be operating in an island of security and its signed data cannot be verified as it cannot send the DS records to the security oblivious parent. Only the root domain authenticates its own data. A zone that is operating in an island of security has to use other means to distribute its trust anchors.

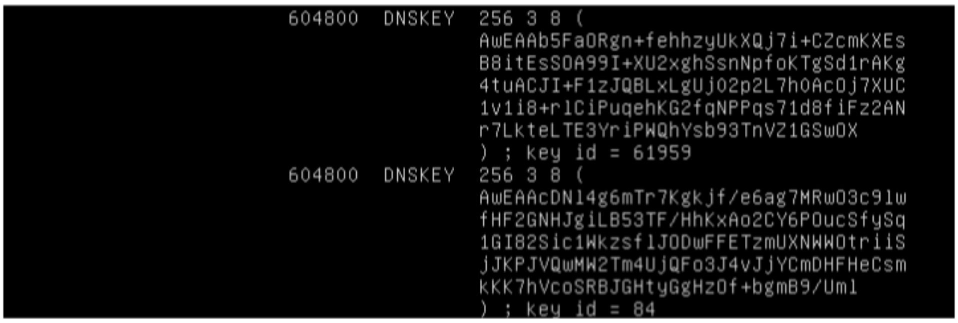
3.2 DNSSEC Additions to the DNS

DNSSEC comes with several changes to the DNS system (Arends *et al.*, 2005a). It introduces new resource records that are used for verifying the integrity and authenticity of DNS data. Message header bits that indicate resolver security awareness and responses with security data are also introduced. The extensions to the DNS also require increases to the normal DNS data packet and increased use of TCP as defined in RFC 2671(Vixie, 1999). All of these extensions to the DNS protocol will be explained in detail in this Section.

3.2.1 Server Side

RFC 4034 (Arends *et al.*, 2005b) details new resource records that DNSSEC adds to the DNS protocol. The standard states that these resources records should only be used to store data pertaining to the DNS only. The resource records that are added to the server side of DNS which consist authoritative name servers are listed and described below:

1. **DNSKEY** - When a zone is signed, the private keys for both the KSK and ZSK are kept away in a secure place and the public keys are stored in the DNSKEY RR (Arends *et al.*, 2005b) to enable resolvers to use them for validating signed data for the zone. The format of the resource record has three fields namely flag, protocol and algorithm number. An example of the DNSKEY RR for demo.com domain used in this research is shown in Figure 3.2.



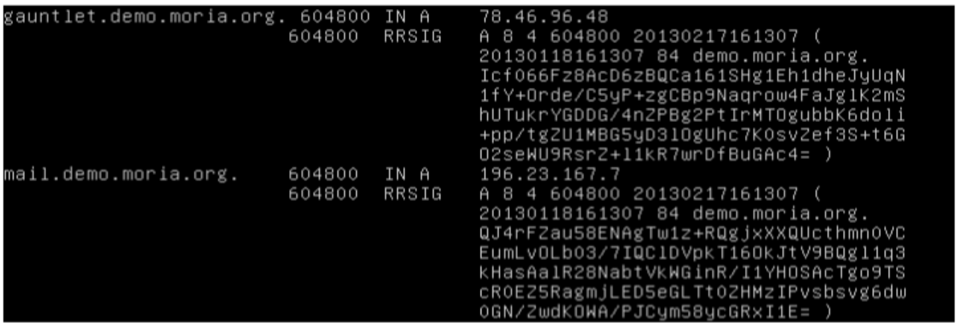
```
604800  DNSKEY  256 3 8 (
                AwEAAb5FaORgn+fehhzyUkXQj7i+CZcmKXEs
                B8itEsSOA99I+XU2xghSsnNpfoKTgSd1rAKg
                4tuACJI+F1zJQBLxLgUjO2p2L7hOAcOj7XUC
                1v1i8+rlCiPuqehKG2fqNPPqs71d8fiFz2AN
                r7LkteLTE3YriPWQhYsb93TnVZ1GSwOX
                ) ; key id = 61959
604800  DNSKEY  256 3 8 (
                AwEAAcDNl4g6mTr7Kgkjf/e6ag7MRwO3c9lw
                fHF2GNHJgiLB53TF/HhKxAo2CY6POucSfySq
                1GI82Sic1WkzsflJODwFFETzmUXNWWOtriiS
                jJKPJVQwMW2Tm4UjQFo3J4vJjYCmDHFHeCsm
                kKK7hVcoSRBJGHtyGgHzOf+bgmB9/Uml
                ) ; key id = 84
```

Figure 3.2: DNSKEY Example

Figure 3.2 shows two DNSKEY resource records for a signed domain. Each DNSKEY resource record starts with the TTL value of 604800 and the resource record type which is DNSKEY. The RR also have flags that differentiate between a ZSK and KSK. Two ZSK in the example above indicated by 256 while 257 identifies a KSK. 3 is a fixed value for the DNSSEC protocol and 8 stands for RSA-SHA 256 algorithm number. The public key present in base64 encoding is in brackets while the key ID value terminates the RR.

2. **Resource Record Signature (RRSIG)** - RRSIG (Arends *et al.*, 2005b) are created by the zone signing software during the signing process and are added to the

signed version of the zone. Their function is to store digital signatures generated by applying the encryption key to the RRs in a zone. The RRSIG are sent together with corresponding unsigned data in response to a query in order to verify whether the unsigned data has not been tempered with. Figure 3.3 shows an example of a RRSIG records.



```
gauntlet.demo.moria.org. 604800 IN A     78.46.96.48
                         604800  RRSIG   A 8 4 604800 20130217161307 (
                                         20130118161307 84 demo.moria.org.
                                         Icf066Fz8AcD6zBQCa161SHg1Eh1dheJyUqN
                                         1fY+Orde/C5yP+zgCBp9Naqrow4FaJglK2mS
                                         hUTukrYGDDG/4nZPBg2PtIrMTOgubbK6doli
                                         +pp/tgZU1MBG5yD31OgUhc7KOsvZef3S+t6G
                                         O2seWU9RsrZ+l1kR7wrDfBuGAc4= )
mail.demo.moria.org.     604800  IN A    196.23.167.7
                         604800  RRSIG   A 8 4 604800 20130217161307 (
                                         20130118161307 84 demo.moria.org.
                                         QJ4rFZau58ENAgTw1z+RQgjxXXQUcthmnOVC
                                         EumLvOLbO3/7IQClDVpkT16OkJtV9BQgl1q3
                                         kHasAalR28NabtVkWGinR/I1YHOSAcTgo9TS
                                         cROEZ5RagmjLED5eGLTtOZHMzIPvsbsvg6dw
                                         OGN/ZwdKOWA/PJCym58ycGRxI1E= )
```

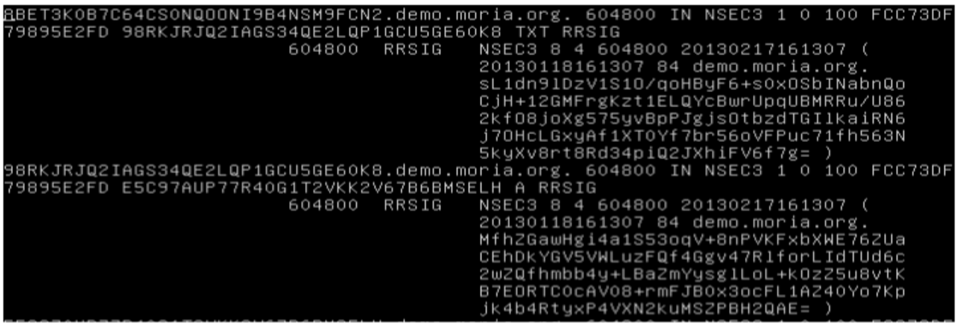
Figure 3.3: Example of RRSIG

The RRSIG shown above are made up of the TTL, RR type, label, algorithm number, type covered, signature expiry date, signature inception date, domain name and the encrypted resource record in that order.

3. **Next Secure Resource Record (NSEC) and (NSEC3)** - NSEC RR are produced when a zone is signed and their function is to certify the non-existence of a RR name between intervals of resource records in a zone (Laurie *et al.*, 2008). NSEC RR are used because denial of existence responses cannot be signed in real time, because encryption keys must be kept off line, as a security requirement. NSEC resource records are created by first listing the zone's names in canonical order and creating a NSEC resource record for each name before signing the NSEC RR with the zone's private key (Yang *et al.*, 2011). The Rdata Section of the NSEC resource records created identifies the name of the RR for which it was created and the next existing name in the zone. If a non-existent name is requested, the name server can send the NSEC RR together with the error message to prove that the requested name does not exist in a secure manner.

   Name servers which use NSEC can give away all the data that is in the domain through a process called zone enumeration (Santcroos and Kolkman, 2007). A collection of all NSEC RR will list all RR names in that zone, thus can reveal registrant data for which domain registrars have an obligation to protect. Zone enumeration attacks can also be used to obtain e-mail address for people associated with the zone for e-mail spamming (Laurie *et al.*, 2008).

NSEC3 is a new resource record introduced to prevent the undesirable NSEC consequence of zone enumeration (Laurie *et al.*, 2008). NSEC3 uses hashes of the zone's names instead of clear text data that is used in NSEC to provide authenticated denial of names in DNSSEC as shown in Figure 3.4 (ISC, 2010). NSEC3 does not completely eliminate zone enumeration, but it does make it more difficult for the attacker, as they will need more processing power to obtain the clear text of the hashes through dictionary attacks (Bau and Mitchell, 2010).

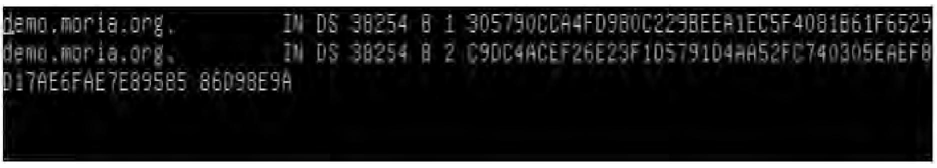


```
ABET3K0B7C64CS0NQ00NI9B4NSM9FCN2.demo.moria.org. 604800 IN NSEC3 1 0 100 FCC73DF
79895E2FD 98RKJRJQ2IAGS34QE2LQP1GCU5GE60K8 TXT RRSIG
                        604800  RRSIG   NSEC3 8 4 604800 20130217161307 (
                                        20130118161307 84 demo.moria.org.
                                        sL1dn9lDzV1S10/qoHByF6+s0x0SbINabnQo
                                        CjH+12GMFrgKzt1ELQYcBwrUpqUBMRRu/U86
                                        2kf08joXg575yvBpPJgjs0tbzdTGIlkaiRN6
                                        j7OHcLGxyAf1XT0Yf7br56oVFPuc71fh563N
                                        5kyXv8rt8Rd34piQ2JXhiFV6f7g= )
98RKJRJQ2IAGS34QE2LQP1GCU5GE60K8.demo.moria.org. 604800 IN NSEC3 1 0 100 FCC73DF
79895E2FD E5C97AUP77R40G1T2VKK2V67B6BMSELH A RRSIG
                        604800  RRSIG   NSEC3 8 4 604800 20130217161307 (
                                        20130118161307 84 demo.moria.org.
                                        MfhZGawHgi4a1S53oqV+8nPVKFxbXWE76ZUa
                                        CEhDkYGV5VWLuzFQf4Ggv47RlforLIdTUd6c
                                        2wZQfhmbb4y+LBaZmYysglLoL+k0zZ5u8vtK
                                        B7E0RTC0cAV08+rmFJB0x3ocFL1AZ40Yo7Kp
                                        jk4b4RtyxP4VXN2kuMSZPBH2QAE= )
```

Figure 3.4: NSEC3 Resource Record

The use of hashes in NSEC3 increases the computational requirements when responding with NXDOMAIN queries and when resolving the same, hence it is recommended that zone owners without non-disclosure requirements use NSEC (Lowinder *et al.*, 2010). TLD domains can use an opt-out option that allows them to use NSEC3 to skip signing zones that have not deployed DNSSEC thereby promoting incremental deployment of DNSSEC (Bau and Mitchell, 2010).

4. **Delegation Signer (DS)** - The DS RR (Gudmundsson, 2003) identifies the key that a child zone uses to sign its DNSKEY RR. This RR is distributed to the parent zone as a way of indicating that the child zone is digitally signed and understands DNSSEC. It is also used in the authentication process of the child zone's resource records. It stores the key tag value, algorithm number and a digest of the DNSKEY RR to efficiently identify the public key of the child domain. The DS RR also has a class, RR type and an algorithm number for the algorithm used to construct the digest. For example, the DS RR for demo.com shown in Figure 3.5 is stored in the .com zone.



```
demo.moria.org.         IN DS 38254 8 1 305790CCA4FD9B0C229BEEA1EC5F4081B61F6529
demo.moria.org.         IN DS 38254 8 2 C9DC4ACEF26E23F1D5791D4AA52FC740305EAEF8
D17AE6FAE7E89585 86D98E9A
```

Figure 3.5: Example of a DS Resource Record

3.2.2 Client Side

The client side of DNSSEC is made up of validating resolvers and end user systems that use DNS data to access Internet services (van Rijswijk-Deij, 2012). DNSSEC also makes changes to validating resolvers which enable them to indicate that they understand DNSSEC. Because of the need for backward compatibility a resolver that is security aware should explicitly indicate so, otherwise name servers will assume that it is security oblivious and not send DNSSEC security data. The changes to facilitate this are described below.

1. **DNSSEC OK (DO) bit** - With the implementation of DNSSEC, there will be security aware resolvers and non-security aware resolvers. Presenting non-security aware resolvers with DNSSEC data will cause an unnecessary processing overhead for them since they are not able to perform signature validation of the digital signatures (Conrad, 2001). Sending signed responses to non-security aware resolvers can cause complete failure in name resolution in the worst case scenario. It is therefore important and necessary for a resolver to indicate that it can process security data so that only security aware resolvers can receive signed resource records for answers to their queries.

   This is achieved by using most significant bit of the Z field on the EDNS0 OPT header on the query. The bit, known as the OK bit, indicates a security aware resolver when its value is 1, whilst a 0 in that field shows that the resolver does not support DNSSEC. DNSSEC aware name servers send DNSSEC data only when they receive a query with a DNSSEC OK bit set. The DNSSEC OK bit should also be copied in a query response. A security aware recursive resolver sets the OK bit even if it is processing a query from a name server that does not support the security extensions, but sends an answer without the DNSSEC resource records to the non-security aware name server.

2. **Message Header bits** - Two message header bits Checking Disabled (CD) and Authenticated Data (AD) are also cited as DNSSEC additions to the DNS. RFC 2065 (Kaufman and Eastlake, 1997) states that the AD bit indicates that the DNS

response has been authenticated by the name server, while the CD bit indicates that the requesting server accepts unsigned data. These two bits occupy preciously unused bits in the DNS query/response format header.

3. **EDNS0** - DNSSEC adds new RR to the DNS thereby increasing the possible size of DNSSEC packets but the packets size allowed for DNS is 512 bytes over UDP. RFC 2671 (Vixie, 1999) makes some changes to these specifications to allow bigger DNS packets and TCP to be used for the DNS. These changes, allows a message of up to 4096 bits to be accepted for the DNS. The changes also facilitates increased message size for other security technologies like TSIG (Vixie *et al.*, 2000).

3.2.3 Domain Look-aside Validation (DLV)

A Domain Look-Aside Validation (DLV) domain is a trust anchor repository that accepts DS resource records for domains whose parents are not signed but want to deploy DNSSEC. Without a signed parent, child domains cannot validate trust anchors of other domains through the normal channel as the chain of trust is broken by the unsigned parent. Such domains are said to be in 'islands of security' (Aitchison, 2005). With a broken chain of trust, a zone (in an island of security) needs to keep trust anchors of all the domains they need to validate. This is difficult, if not impossible to put in practice. A solution to that problem is the use of DLV domains that collect trust anchors of other domains so that they can be used for security validations by zones in islands of security (Weiler, 2007).

As already alluded to, the root domain was signed in 2010 but a significant number of other TLD's have not been signed making the continued existence of DLV domains necessary. The SecSpider DNSSEC monitoring project (SecSpider, 2012) recorded a substantial increase in the number of signed zones after signing of the root domain. Some of these zones operate in islands of security and use DLV domains to enable them to publish their trust anchors and be able to validate DNSSEC data.

In the example in Figure 3.6, *dlv.example.net* is the DLV domain. NS1 is configured with the trust anchors of dlv.example.net and the target domain example.com has its trust anchors in the DLV domain because its TLD, .com is not signed. A security aware NS1, first queries the root domain for example.com trust anchors through the normal channel (1). The trust anchors will not be found because there is broken chain of trust at .com. It will use DLV domain (2) where they can be obtained and validated. The security oblivious NS2 will perform the usual name resolution without security data

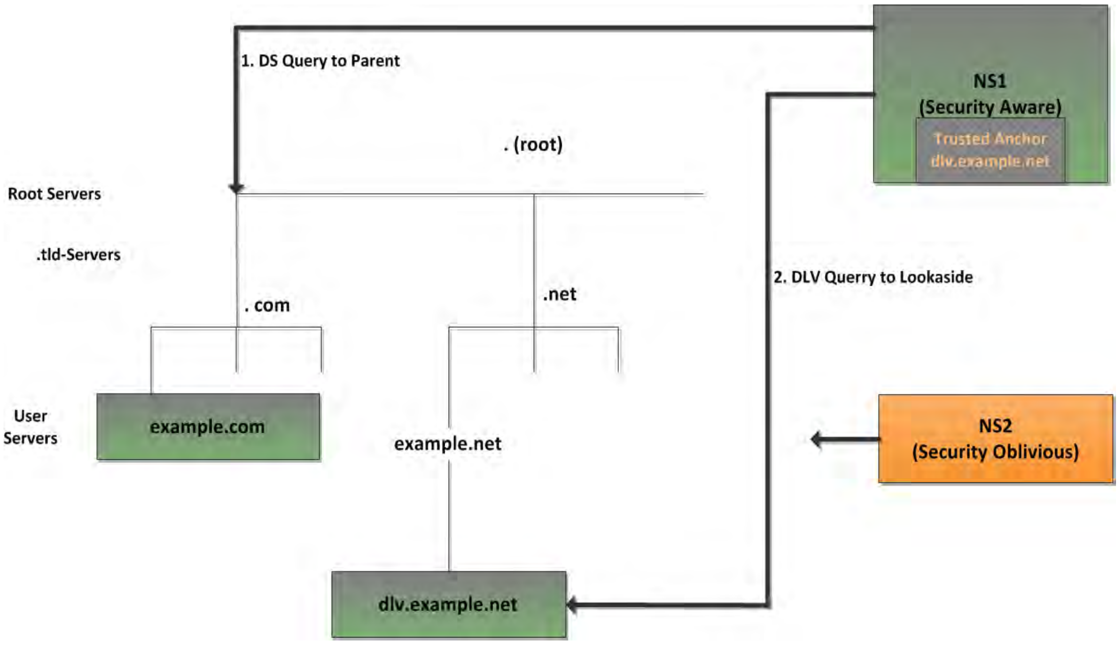


Figure 3.6: DLV Verification (Aitchison, 2005)

Four trust anchor repositories that can be used as DLV domains are listed on DNSSEC Deployment Initiative website (DNSSEC Deployment Initiative, 2012). The two that are still functioning at the time of writing, are ISC DLV Registry[1] and Sec Spider [2].

3.2.4 DNSSEC Encryption Algorithms

DNSSEC allows specific data encryption algorithms to be used for message encryption in securing DNS traffic (IANA, 2012a). Each algorithm usable for message encryption in DNSSEC has a value that is used to uniquely identify it in the DNSSEC resource records, which are DNSKEY, DS and RRSIG. The value is stored in the algorithm number field in the RDATA Section of the resource record.

The algorithm number values have been set to take values 0-255 but currently values 15-122 are unassigned, whilst 0, 123-251 and 255 are reserved and values 253 and 254 are used when an entity uses private encryption algorithms (IANA, 2012a). Other algorithm number values are not used in DNSSEC, but in other DNS transaction security mechanisms that include TSIG and SIG(0). These include values 1 and 2 which stand for RSA/MD5 and Daffie-Hellman. Some of the algorithms can be used for both DNSSEC and transaction security and these are DSA/SHA1 and RSA/SHA-1. The IANA website (IANA, 2012a) shows the algorithms currently in use at the moment.

[1] https://dlv.isc.org/

[2] http://secspider.cs.ucla.edu/

3.3 Key Management

DNSSEC uses two encryption keys namely the zone signing key (ZSK) and the key signing key (KSK) (Aitchison, 2005). Each key is made up of private and public keys. The ZSK is used to sign the resource records of the zone while the KSK is used to sign the ZSK (specifically the DNSKEY RR) to create a secure entry point for the zone. It is recommended that KSK be stronger, as they are used externally to verify data from the zone, and they usually have a longer life span (Lowinder *et al.*, 2010). The security of the signed domain is heavily dependent on these encryption keys. Actually, these encryption keys are the hallmark of the security extensions to the domain system. Thus, if compromised, there will not be any security. It is therefore important to take every step possible to secure the encryption keys, especially the private key components. The process of ensuring security of these keys brings with it administration overheads to systems administrators managing domains.

Cryptographic systems security is based on using an encryption key for a period that is shorter than the time that it will take to break that key through cryptanalysis using current computational power (Schneier, 1996). Length of the keys increases the system's security as longer keys require more computational power and time to break. This means that, a longer key with a shorter life span will provide more security but will also require more computational power from the legitimate system using it. A shorter key lifetime will also increase administration overhead in changing it regularly. It is therefore important to strike a balance between the level of security required and the costs involved in terms of computational power and administration requirements.

3.3.1 Key Generation and Storage

The environment in which the encryption keys used in DNSSEC are generated should be secure enough to prevent unauthorized access by entities with malicious intentions. RFC 4641 (Kolkman and Gieben, 2006) recommends that the computers on which these keys are generated should not be connected to the network or special hardware devices should be used to generate the keys to ensure security. Other important aspects of key generation cited are randomness of the key generator and choice of the algorithm. If the key generating pattern can be predicted or replayed, the keys can be compromised. It is recommended, therefore, that the key generator should follow the guidelines detailed in RFC 4086 (Eastlake *et al.*, 2005) to ensure true randomness of the key generation process. RFC 4641 (Kolkman and Gieben, 2006) together with NIST (Chandramouli and Rose,

2010) and ENISA (Lowinder *et al.*, 2010) guidelines lists three algorithms that can be used for DNSSEC and these are RSA, DSA and Elliptic curve. All the documents, however, recommend RSA-SHA1 for key generation, as it offers better performance in validation than DSA. The three guidelines further recommend use of RSA-SHA-256 for DNSSEC.

A DNS and DNSSEC analysis paper for Sweden (Lowinder, 2012) shows that by early 2012, RSA-SHA-256 was the most used algorithm, with its usage exceeding 500 000. The reasons given for wider use of this algorithm is that SHA-256 is more secure than SHA-1 since it is less likely to produce the same hash. RSA-SHA-1 was the second most used algorithm and was used for just over 50 000 domains. The paper discourages usage of DSA which was used by only 1 domain on the basis that it requires more CPU for its signature validation and also because it requires a good random number selection system. The Sec Spider project (SecSpider, 2012) website which monitors DNSSEC also shows that RSA-SHA-1 is the most used algorithm.

Storage of the private key portions of the ZSK and the KSK is equally as important as key generation in that if the keys are not securely stored, they can be obtained by adversaries, thus putting the domain's security at risk. RFC 4641 (Kolkman and Gieben, 2006) recommends that the keys together with the master copy of the zone storage should have strong physical and logical access controls to prevent unauthorized access. It is also recommended that in addition to storing the keys off line, fault-tolerant storage media should be used for back up of the private key and zone files (Chandramouli and Rose, 2010).

In instances where dynamic updates are used, the private keys and master file are required to be available online. In such a scenario, the recommendation is to use a hidden master, i.e the master server must be invisible to external hosts but visible to secondary name severs in the network. Security in such instances is purely a function of using the correct technical setup and good system administration practices. Security technologies such as TSIG (Vixie *et al.*, 2000) and IPSec (Kent and Atkinson, 1998) can be used to ensure security between the name servers involved.

3.3.2 Key length and Validity Period

The greater the time an encryption key is used the greater the risk that it can be compromised through various methods that include cryptanalysis, espionage, carelessness or accidents (Schneier, 1996). An encryption key's validity period in DNSSEC should be reasonable to minimize compromise of the key. Further, to minimize administration overheard, frequent key rollovers should be avoided. RFC 4641 (Kolkman and Gieben, 2006) recommends that a ZSK should be changed after 30 days when its 1024 bits and a KSK should be changed after 12 months when its length is 2048. NIST (Chandramouli and Rose, 2010) however differs when it indicates that at the same length, the ZSK can be used for up to 90 days and KSK up to 24 months.

Research by Massey *et al.* (2007) of Sec Spider reported that most DNSSEC tools come with a default key validity period of 30 days, and that early DNSSEC adopters accepted the default key validity period. To improve security, it is advised that the key lifetime period should be varied so as to prevent prediction of when the process will happen. The recommendation by Chandramouli and Rose (2010) of using a wider range of key length can be useful for this purpose because it will allow more random key rollover periods. Security Week Network website (Mohan, 2010) suggests that rolling over the ZSK key either 10 days late or earlier helps reduce prediction of key rollover.

In a research paper on DNSSEC key management Kos *et al.* (2012) indicates that a key with a bigger length takes longer to generate. This reinforces the argument that the bigger the size of the key, the more computational power required. Different algorithms may require different computational resources as reported by NIST (Chandramouli and Rose, 2010).

The Sweden report (Lowinder, 2012) shows that the longest key length is 4096 bits for the KSK, although most KSK are 2048 bits in length. The shortest KSK which is explicitly discouraged in the report is 512 bits. The most popular key size for the ZSK in the same research paper is 1024 bits, followed by 512 bits and 2048 bits. A key length of 1536 bits also has a significant number of users. The paper however concludes that the KSK is generally longer than the ZSK.

3.3.3 Key Rollover

As already explained in the previous Section, cryptographic systems require that the encryption keys should be changed before an attacker can have enough time to break the key through such attacks as brute force and cryptanalysis. The IETF RFC 4641 (Kolkman and Gieben, 2006) explains that encryption keys should be changed on or before the date that they expire because once they expire they cannot be used to validate DNS data and the affected domain becomes unavailable. The period of unavailability can become longer as name servers often provide caching services. The expired keys could be cached for long periods depending on the RR's TTL. The process of changing the keys is termed key rollover in DNSSEC.

A zone's TTL should never be equal to the signature expiry date. The standard recommends that a zone's maximum TTL should be shorter than the key validity period to allow renewal of cached records before signature expiry. This will avoid updating of expired keys and resource records at the same time.This will allow key rollover when the authoritative name server's load is not at its peak. Re-signing a zone just before the signature validity period expires can cause simultaneous expiry of cached records thereby resulting in peak loads for authoritative name servers.

Another recommendation put forward involves the minimum TTL for the zone. This should be long enough to allow recursive name servers to benefit from caching as well as to avoid records expiration before the process of validity is complete. Zone resigning should be done earlier before signature validity expiry so that slave servers for the zone can have ample time to do zone updates before the RR's signatures expire. It is therefore recommended that a zone's SOA validity period be at least a quarter of the signature validity period so that problems with zone updates can be noticed in advance. Programs that notify system administrators of impending signature expiration for slave servers are suggested.

Keys that have been changed should also be kept in the zone for some time so that they can be used to validate cached data that has the old keys. Removing these keys from the zone on key rollover can result in unavailability of the zone for some clients.

There are two methods that can be used to change keys for signed domains and these are the pre-publish and the double signing methods. Key rollover for the ZSK is simpler as it does not involve the parent zone and changing of the trust anchors whereas KSK rollover involves communicating with the parent zone and changing the DS resource record that is kept in the parent domain's zone files. A choice can be made between the two methods when performing key rollover for the ZSK but only the double signature method can be used for the KSK. A simplified process of doing the rollover using these two methods is presented in a publication on Unix and Linux Administration (Whaley *et al.*, 2010)

1. **Pre-publish** - With this method, the key is included in the zone before it is used is to sign the zone. The purpose is to publish it to other name servers so that it is cached well before it is used (Aitchison, 2005). The advantage of this method over double signing method is that it does not increase the size of the zone especially when used for the ZSK. This method can become handy in instances where the key has been compromised as the old key can be dropped and the new key immediately used to sign the zone, given its already known by other security aware name servers. The disadvantage for this method is that the new key is published early and this

increases the time it is exposed to risk of cryptanalysis or brute force attacks.

2. **Double signing** - Double signing involves signing a zone with two keys at the same time. It ensures that both the old and the new key can be used to validate the zone's data. Further, it ensures that old keys can be deleted when all the caching name servers are in possession of the new key. If this method is used for ZSK rollover it results in an increase in the size of the zone as there will be a dual signing of the resource records. This significantly increase the size of the zone, as such, it may not be acceptable for some large zones. Both Aitchison (2005) and Whaley *et al.* (2010) recommend the use of this method for KSK rollover, as this will result in an insignificant increase in the size of the zone, since the KSK sign the DNSKEY RR only.

3.4 DNSSEC Deployment

In DNSSEC, the validation of a child domain's signed data is done by the parent zone (Massey *et al.*, 2007). After signing its zone, the child zone distributes its trust anchors to the parent zone for use in validation of its resource records. Each parent domain validates the resources of its child domain except for the root domain which validates for itself. This means that a child zone needs its parent to be signed for its signed data to be validated (Yang *et al.*, 2011). A signed child domain whose parent is not signed is referred to as an island of security and will need to make use of trust anchor repositories to distribute its public keys. It is therefore necessary for the root domain and all the TLD's to deploy DNSSEC. The next Section will look at the progress made so far in the deployment of DNSSEC.

3.4.1 Progress in DNSSEC Deployment

The ICANN DNSSEC status update report (ICANN, 2012f) shows that the root domain was signed on 15 July 2010 after recording positive results from a series of tests to determine the performance of the signed root. The KSK used was 2048 bit RSA whilst 1048 bit RSA was used for the ZSK and the algorithm used was RSA/SHA-256. The report also states that there were no problems reported when the root zone was signed and that nine TLDs DS RR were uploaded onto the root by 10 July 2010.

The European Union report on DNSSEC deployment worldwide reports that when the root zone was signed on 15 July 2010 only 32 TLD's were signed (EURid, 2010). The signing of the root domain has however resulted in an increase in the total number of TLD's signed as revealed by statistics on the ICANN website (ICANN, 2012f). As at 23 October 2012, the website gave statistics that show out of the 314 TLD's in existence, 102 of them have been signed. This represents 32.5 % of the registered TLD's. Out of the 102 TLD's that are signed, 94 have their trust anchors in the root domain while 3 also distributed their trust anchors to ISC's DLV. Figure 3.7 obtained from the DNS-OARC website[3] shows the count and percentage of TLD's with DS records in the root at the time of writing.

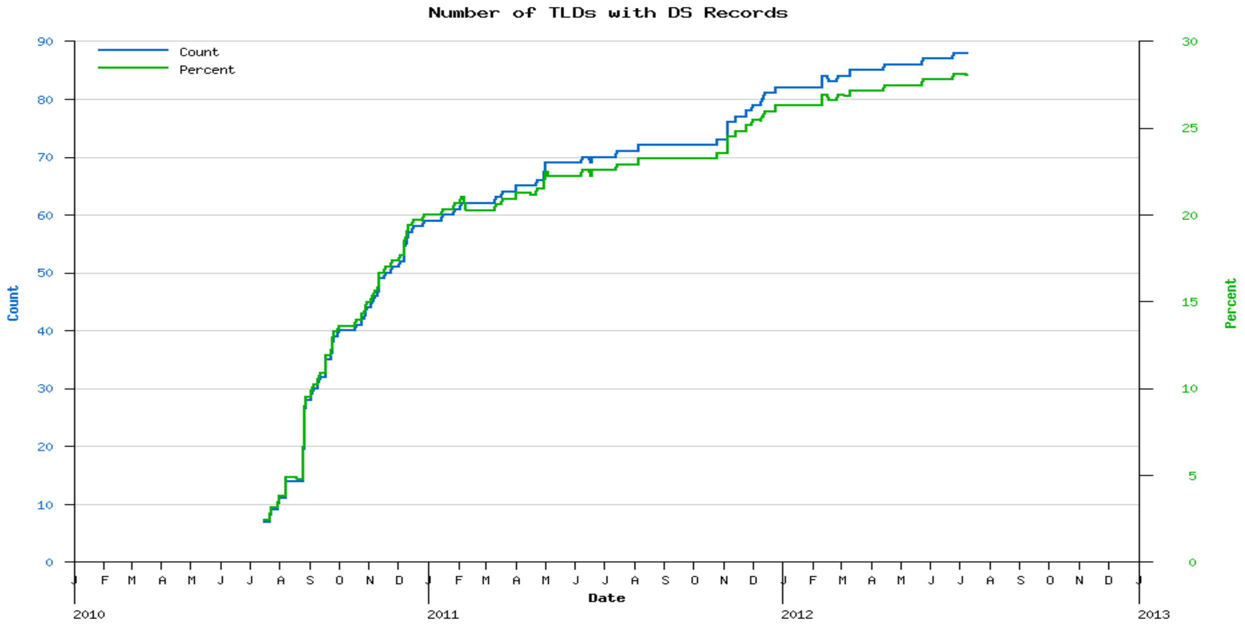


Figure 3.7: Number of TLD's with DS Records (DNS-OARC, 2012c)

Out of the 98 signed TLD's, only 2 ccTLD's from Africa are signed and these are .ug for Uganda and .na for Namibia signed on 01 September 2009 (DNSSEC Deployment Initiative, 2009) and 01 October 2010 respectively. The map in Figure 3.6 below showing DNSSEC deployment patterns in the world illustrates confirms this report. A Research by Huston (2012) put the number of DNSSEC validating resolvers at 4%. Libya is the only African country in the top 10 of countries with the most DNSSEC validating resolvers. The research estimates that 9% of end user hosts perform DNSSEC validation (Huston, 2012).

Other countries like the US has gone a step further by making it mandatory for all government departments to deploy DNSSEC (Rasmussen, 2011). The graph in Figure 3.8

[3] https://www.dns-oarc.net/oarc/data/zfr/root/ds

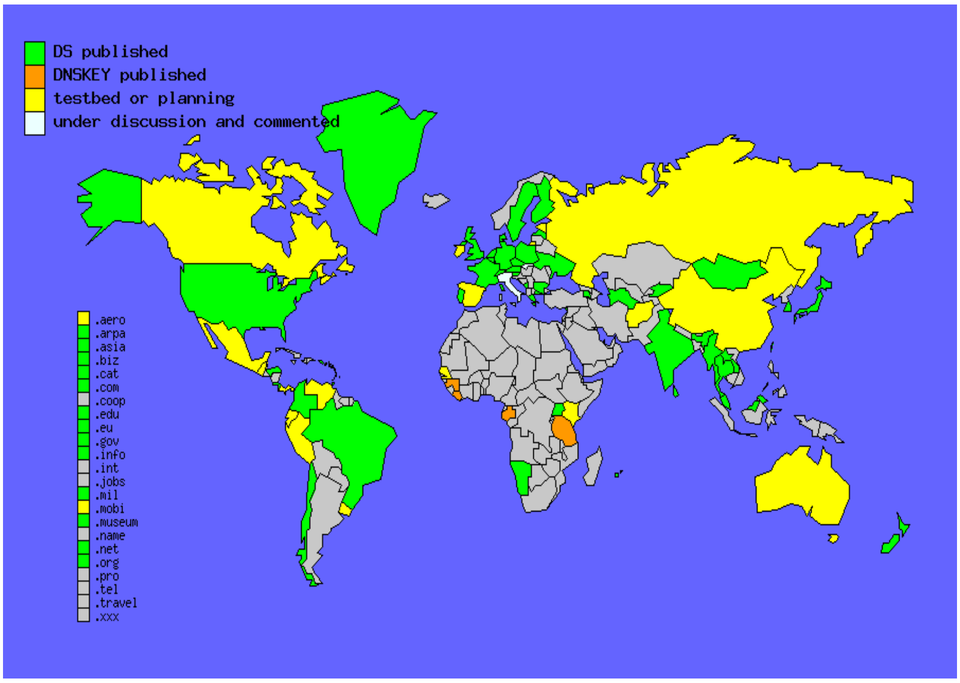


Figure 3.8: DNSSEC Deployment Map as at 30/11/2012(OHMO, 2012)

below obtained from Sec Spider DNSSEC monitoring project (SecSpider, 2012) shows a sharp increase in the number of DNSSEC enabled zones since 2010 and the growth pattern is continuing on an upward trend. The graph indicates that even though less than half of the TLD's are signed, much more lower level domains are signed.

Some countries have shown tremendous progress in deployment of DNSSEC as can be gathered from the presentations at the FOSE 2012 conference (DNSSEC Deployment Initiative, 2012). A US government policy enforcing mandatory deployment of DNSSEC by all government departments resulted in an increase in the percentage of compliant domains from less than 40% in July 2011 up to 55 % in March 2012 (Stoyanov, 2012). A .se domain report on DNSSEC deployment in Sweden (Lowinder, 2012) show that out of a total of 1195719 registered domains, 174 487 domains are signed. Peter Janssen (2012) of .eu domain presentation at ICANN 44 show that there has been a sharp rise in signed domains in the year 2012. A report by PowerDNS (2012) claiming that Netherlands has the highest number of signed zones states that 1331987 zones have been signed.

Software development has not lagged behind as far as DNSSEC is concerned as shown on the DNSSEC deployment initiative website (Dnssec Deployment Initiative, 2012). DNS server software identified on the website include Bind, NSD, Unbound, ANS, CNS and KnotDns. Windows Server 2008 R2 and its client version Windows 7 also support DNSSEC (Seshadri, 2008) but van Rijswijk-Deij (2012) recommends that organizations

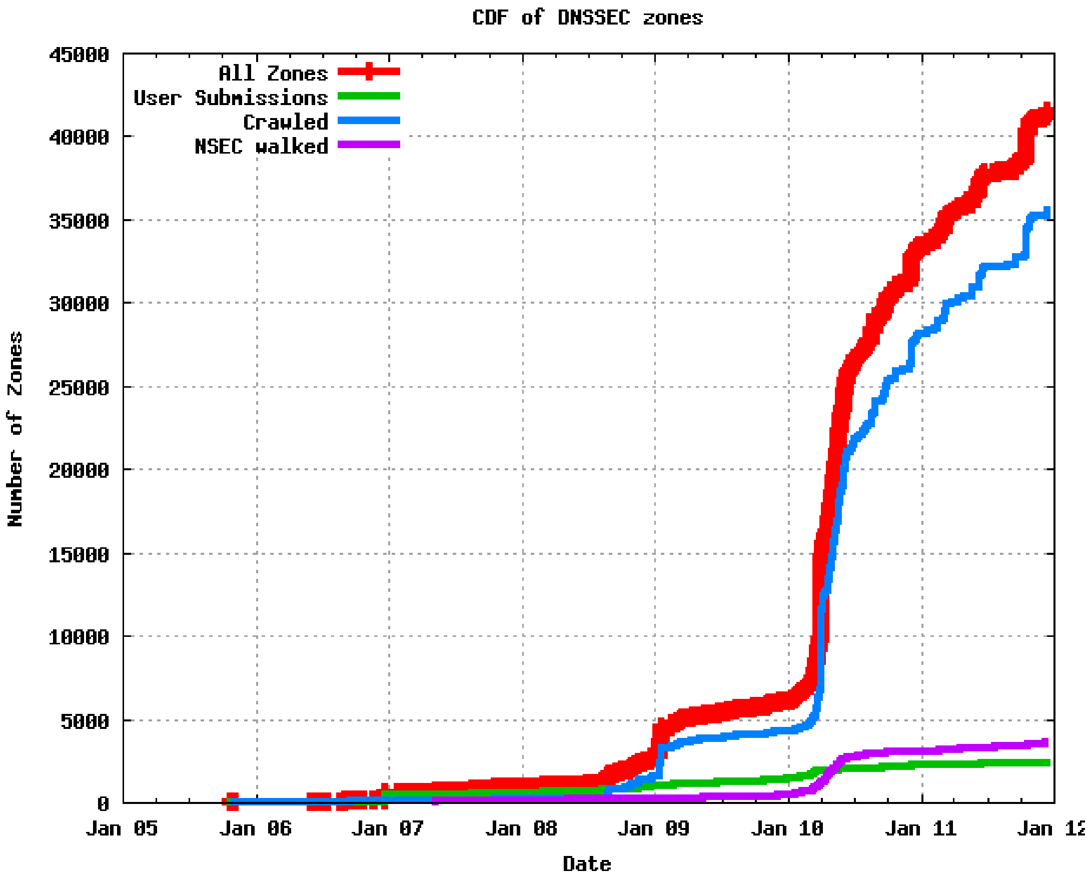


Figure 3.9: DNSSEC Signed Zones (SecSpider, 2012)

should use the latest version which fully implements DNSSEC features and automates most signing processes.

Also listed on the website (Dnssec Deployment Initiative, 2012) are DNSSEC key management tools, most of which are open source and come bundled with open source DNSSEC name servers like Bind. These tools include key generation and rollover tools as well as zone signing tools like donuts, Rollerd, dnssec-zonesigner etc. Other DNSSEC resources listed include are hardware modules, zone monitoring and troubleshooting tools, parent zone tools for DS updates and DLV domains.

The site also shows a list of end user DNSSEC enabled applications like Firefox Add-on and DNSSEC-Trigger. DNSSEC development libraries which can be used to develop DNSSEC enabled applications are also summarized. The websites also gives links to useful tutorials and guides that can be helpful when deploying DNSSEC.

3.5 DNSSEC Deployment Scenarios

The situation in which the world finds itself in today in terms of deploying DNSSEC is shown on the matrix below. Now that the root domain is signed, every domain finds itself with either a signed or an unsigned TLD. The steps needed to adopt DNSSEC in those circumstances are different for two different sets of users which are users within an organization accessing the Internet from LAN and external Internet clients accessing the Internet through their ISP. The matrix below shows the situation in which these two groups of users find themselves in today as far as DNSSEC is concerned.

Table 3.1: DNSSEC Deployment Matrix

	Signed Root	Island of Security
Internet Client	A (Signed root, signed ISP)	B (Signed root, unsigned ISP)
Intranet Client	C (Signed root, signed TLD)	D (Signed root, unsigned TLD)

3.5.1 Internet Clients (A and B)

These are users who do not have their own name servers and access the Internet via their ISP's name servers. Their stub resolvers depend on the the resolvers of their ISP's for name resolution and they cannot deploy DNSSEC on their own (York, 2012). ISP's therefore need to deploy DNSSEC for this class of users to get validated DNS responses. If the ISP is signed as is the case with all ISP's in Czech Republic and Sweden, they get validated responses. These Internet clients may be faced with a situation where their ISP is not signed and cannot validate security data (van Rijswijk-Deij, 2012). In that case they can use other mechanisms such as local resolvers and validating applications to verify DNS responses they get from other name servers.

Sweden and Czech Republic have made a lot of progress in this area as all the ISP's in the country are reported to be providing DNSSEC services to their clients. Czech Republic reports (Filip, 2012) that 36.5% of registered .cz domains have been signed and the numbers are growing. The US also reports that their major domain registrars like GoDaddy, Dyn and GKG support DNSSEC (NIST, 2012). The ICANN website (ICANN, 2012d) showed 17 domain registrars that support DNSSEC as at 19 April 2012. None of these domain registrars is from Africa.

3.5.2 Intranet Clients (C and D)

Intranet clients use their organization's name servers for name resolution and do not depend on ISP's for name resolution. The organization however may find itself with a signed or an unsigned TLD. When the TLD is signed the organization will need to deploy DNSSEC for their local name server and distribute their trust anchors to the parent domain so that their internal users can have security validated responses (Internet Society, 2012). In this instance deployment of DNSSEC is totally in the hands of the organization. If the organization does not sign the domain, users can also use local and public validating resolvers.

When the TLD is not signed, the organization can still deploy DNSSEC and use DLV domains for distributing their trust anchors so that their users can have secured name resolution (ISC, 2009b). This is the most common position for domain name operators in Africa as most countries have not signed their TLD's. A large organization operating a WAN can sign its domain and other sub domains it operates so that its users can have authenticated DNS responses for local queries.

3.6 Incentives for Deploying DNSSEC

DNS security threats such as cache poisoning and packet interception pose a great security challenge to business service providers (ENISA, 2009). When users are redirected to a forged domain, other attacks such as phishing occur resulting in financial losses and bad reputation for the victim organization (Dagon *et al.*, 2008). Individuals can also be victims of identity theft and fraud in the DNS security attacks while organizations can have their communications intercepted in espionage attacks (van Rijswijk-Deij, 2012).

A signed domain cannot be subjected to DNS attacks outlined in Chapter two because validation checks done by security aware resolvers can detect and drop responses that have been forged or are coming from malicious sources (EDUCAUSE, 2010). Therefore, organizations that sign their domains can prevent these DNS attacks against their domain name and prevent financial losses and risks of bad reputation that come as a result of the attacks (ICANN, 2012a). Internal use of DNSSEC will also secure name resolution process especially for large organizations with remote offices connected by WANs that can be subject to DNS attacks.

With the low rates of DNSSEC deployment in most countries, early adopters of the technology stand to benefit from business competitive advantages against their competitors who are still to deploy the technology (Lowinder *et al.*, 2010). DNSSEC can improve security for sensitive online services as internet banking by ensuring that clients connect to the correct domain before TLS/SSL can be used to secure the client-server communication (Eland, 2009). Companies with signed domains can argue that they used the best available technology to secure the name lookup process in legal issues associated with DNS attacks (Lowinder *et al.*, 2010).

DNSSEC suffers what is referred to as the chicken-egg problem by van Rijswijk-Deij (2012) in that people are reluctant to sign their domains if there is no one who is going to validate their signatures. For example a local bank will not see the benefit of deploying DNSSEC if local ISP's do not have validating resolvers because no one is going to use the added security to their Internet banking. Countries should therefore have their TLD's signed so that it becomes easy for ISP's to sign their domains and help improve internet security for their countries (van Rijswijk-Deij, 2012).

DNSSEC is set to be used to provide greater security for the internet security in the future (Murali, 2010). There are suggestions that it can be used to distribute SSL certificates and also help improve the security of other security technologies such as IPSEC and SSH (Pokai Chen, 2010). There are also calls for the development of DNSSEC validating applications that can provide a solution for the security challenges that exist between a validating resolver and a stub resolver (Hardaker and Krishnaswamy, 2012). With the future cyber security most likely to be based on DNSSEC, it is important that ccTLD's sign their zones to encourage organization in their country to do the same in preparation for the future (ICANN, 2012a).

3.7 DNSSEC in the future

While DNSSEC is critical to providing authentication and integrity to the name resolution process, the security features it provides can also be used to improve security of other protocols such as SSH as explained below. The protocol also presents a cheaper and secure alternative to Certificate Authorities in the PKI. This Section highlights how SSH and SSL/TLS can be integrated into DNSSEC in the future.

3.7.1 SSH and DNSSEC

SSH is a protocol that uses asymmetric key cryptographic systems to authenticate and secure communications with remote hosts (Schlyter and Griffin, 2006). The SSH server uses the private key for data encryption and sends a hash of the corresponding public key to the SSH client for verifying the public key. The hash of the public key is sent to the client when a connection is established between the client and the server. The client will need to verify the public key with the hash before using it for securing the communication channel. On the first instance of receiving the hash, out of band means such as phone calls and e-mails can be used to verify the hash.

SSH design overlooked the need to bind the public key hash to the server for clients who cannot use out of band means to verify the authenticity of the hash received when a connection to the server is initiated (Ali and Smith, 2004). In such situations, the SSH client is vulnerable to man MITM attacks and risks losing their login details to an attacker who impersonates the SSH server and supply a forged hash. The DNS can be used to distribute the SSH server public key hash thereby providing a way of authenticating the hash.

The security of this remote server login protocol largely depends on the verification of the public key fingerprint. Out of band means are not always done and people end up trusting the hash they receive (Whaley *et al.*, 2010). Whilst the DNS can provide a solution to this problem it is subject to security attacks detailed in Chapter 2 and cannot be relied on for fingerprint verification. However the security extension to DNS can provide the required security as it provides a way of checking for integrity and authenticating the SSHFP resource record (Schlyter and Griffin, 2006) which contains the hash.

DNSSEC therefore provides security for the SSH fingerprint distribution to clients (Hardaker and Krishnaswamy, 2012). However, if the SSH client cannot do DNSSEC validation itself it has to employ transaction security mechanisms such as TSIG between itself and its name server for security (Schlyter and Griffin, 2006).

3.7.2 TLS and DNSSEC

Transport Layer Security (TLS) certificates are used to authenticate and encrypt communication between a web server and a web browser (Dierks and Rescorla, 2008). The web server has a private key and it sends a digital certificate containing the corresponding public key to the web browser. The key pairs are used for data encryption and decryption by the communicating parties so that data that is exchanged can be authenticated and its integrity can be verified. The web browser trusts the digital certificate from the server because the certificate is issued by a trusted entity called the Certificate Authority (CA) which it believes verifies identities before issuing certificates (Pokai Chen, 2010).

Whilst TLS provides authenticity and integrity for the communication between the web server and the client, DNSSEC provides the same for the name resolution process (Eland, 2009). The two security technologies will complement each other in that DNSSEC will verify the domain and TLS will secure packets that are exchanged between the web server and client. For example when a client types www.ru.ac.za in their web browser, DNSSEC will ensure they connect to the domain ru.ac.za and TLS will encrypt the data exchanged between them and the web server thereby providing integrity and authenticity.

The DANE working group is however working on ways of using DNSSEC to distribute TLS Certificates such that the expensive role of CA's will be abolished (Barnes, 2011). CA's have been compromised, their certificates stolen and used in phishing attacks. The proposed development that DNSSEC will store digital certificates as resource records will provide a way of authenticating the source of digital certificates before they are used in the communication channel.

3.7.3 DNSSEC Validating Applications

Most end user computers use a stub resolver which depend on an external validating resolver for name resolution (Mundy, 2012). There is a risk that domain names that have been validated can be spoofed when the response travels from the resolver to the client. This can happen on both computers using a resolver supplied by their ISP and those using a resolver in a local network.

Although the network path between the recursive server and the end user can be secured by other means such as IPSEC, TSIG and other network layer techniques, this is seen as adding complexity in the validation process. Use of transactional security mechanisms for this purpose may not be practical to implement because a resolver usually serves a big number of client computers (Vixie *et al.*, 2000).

It has been suggested that building applications that can do DNSSEC validation will deal with the security problem for the communication channel between the client computer and the resolver (Hardaker and Krishnaswamy, 2012). A DNSSEC validating application will detect if DNS data received from the resolver has been modified. Some API's have been developed for several programming languages to encourage programmers to develop DNSSEC aware applications. It has been noted that even though the process is difficult, the benefits justify the effort.

DNSSEC validating applications will also provide useful error message like the Firefox Add-on which specifically tells an end user that the site they are visiting is not secure

(Mundy, 2012). An application that can perform DNSSEC validation offers security to users regardless of their location.

3.8 Experiences with DNSSEC

Although DNSSEC provides much needed security to the domain name system, it comes with its own weaknesses as highlighted in RFC 3833 (Atkins and Austein, 2004). When the RFC was published the root domain was not yet signed and this was noted as a major obstacle in the deployment of DNSSEC. Now that the root has been signed and a significant number of both generic and country code TLD's are signed, this obstacle has been overcome and this proves that the world is slowly moving close to total security of the DNS.

The deployment of DNSSEC still remains a challenge in most African countries as can be seen from the DNSSEC deployment map (ICANN, 2012c) that most African countries like Zimbabwe and South Africa still have their domains unsigned with no reports of progress in doing so. This presents a challenge to domains using these ccTLD's as they will need to use DLV domains for publishing their trust anchors if they are to sign their domains.

DNSSEC increases the size of DNS packets significantly and this was noted as a factor that increases DNS vulnerability to denial of service attacks (Santcroos and Kolkman, 2007). As explained in previous sections DNSSEC has made it necessary to increase the allowed DNS message size from 512 to 4096 bytes and open TCP for DNS messages (Vixie, 1999). The increase in DNS packets also increases the workload of resolvers as it has to process more data and also perform signature validation (Bau and Mitchell, 2010). The increased message processing time will also result in an increase in the time it takes a resolver to provide an answer to a client. This has negatively affected people's perceptions about DNSSEC and may be the cause of the slow adoption of the protocol extension.

DanYork (2012) of the Internet Society summarized the challenges they discovered while they were setting up the Deploy360 web portal for DNSSEC deployment[4]. They classified the challenges into three groups namely domain consumers, domain owners and domain name infrastructure operators challenges. The domain name owners were defined as the end users of the DNS whilst domain owners are entities who register their own domains and want to deploy DNSSEC. Lastly domain infrastructure operators include domain registrars, domain hosting companies and ISP's challenges. The challenges encountered by these three groups according to this publication are summarized below.

[4] http://www.internetsociety.org/deploy360/dnssec/

3.8.1 Domain Consumers Challenges

A challenge with DNSSEC deployment for end users is that while a variety of tools have been developed for systems and network administrators, there are only a few applications to date that can do DNSSEC validation for end users. The few applications noted so far are the web browser plug-ins for Firefox, Internet Explorer and Google Chrome (Dnssec Deployment Initiative, 2012). The plug-ins for these web browsers need the end user to actually do the installation themselves and users without technical knowledge or who do not know about DNSSEC will most likely not make use of these applications.

Another challenge noted for end users is that there is no standard way defined so far that can be used to let people know whether the site they are accessing is signed and whether the signatures have been validated. The paper recommends that a standard way of informing application users of DNSSEC applications should be identified by the DNSSEC community. It is also highlighted that most end users use recursive name servers supplied by their ISP's but most ISP's have not implemented DNSSEC. This means that in as much as the end user may want DNSSEC, they are not able to use it. Other methods of using DNSSEC that include end client validating resolvers like DNSSEC-Trigger (NLnetLabs, 2012b), public security aware name servers like OARC (DNS-OARC, 2011) require technical know-how thereby disadvantaging the average end user.

3.8.2 Domain Name Holders

For owners of registered domain names who want DNSSEC, the problem is once again ISP's who do not support the DNS protocol extension. The author suggests that if all ISP's, domain hosting companies and domain registrars could automatically sign all their clients domains like Binero of Sweden, it will be easy for domain name holders to deploy DNSSEC.

The author also explains that the separation of the DNS hosting and registration functions also present difficulties in that the hosting company need to maintain and manage the DNSSEC operations whilst trust anchors need to be distributed to the domain registrar and in some instances one of the two may not support DNSSEC. If the two functions could always be done by one entity, it will be much easier for domain owners. The setup of some organizations where they can have different websites hosted by different service providers and mail services also hosted by external service providers can result in complications especially where some of the service providers do not support DNSSEC.

3.8.3 Domain Name Infrastructure Operators

The author argues that domain name infrastructure operators can be aware of the currently available DNSSEC documentation but they need more detailed material that can help them better understand the whole processes and steps involved in setting up DNSSEC. Setting up a DNSSEC environment that can automatically sign a client's domain and perform such processes like key management and automatic upload of trust anchors requires much technical know-how and resources. Processes like key management require automation as manual process are prone to errors thereby increasing the stakes for the typical domain name service provider.

3.8.4 Politics

Some countries such as Russia and China are reluctant to deploy DNSSEC as they are not happy that the DNS system is administered by the US (Whaley *et al.*, 2010). Russia are reported to have refused to deploy DNSSEC insisting on their own symmetric algorithm called Ghost when they know that DNSSEC uses asymmetric encryption algorithms. It is also highlighted that other countries like China have their own root .com and .net servers and are not wiling to deploy DNSSEC.

3.9 Alternative Approaches to DNS Security

This Section looks at alternative ways of providing security to the domain name system. A brief outline and critical analysis of each methods is presented below.

3.9.1 DNSCurve

DNSCurve Project website (Timmers, 2009) defines DNSCurve as a way of improving DNS security by adding integrity, availability and confidentiality through the use of elliptic-curve cryptography on DNS data. The encryption is done at the link level layer of the TCP/IP protocol suite and is designed to work with the current firewall configurations. Elliptic-curve algorithms are defined as more recent, faster and secure than RSA algorithms commonly used for DNSSEC implementations (Edge, 2009). This is taken to mean that DNSCurve is more secure as it takes more time and resources to break the elliptic-curve algorithm through brute force attacks as compared to RSA.

Daniel Bernstein (2010) the developer of DNSCurve argues that DNSCurve sends encrypted DNS data packets thereby providing confidentiality which DNSSEC does not provide. He also demonstrated that DNSSEC can be used as a tool to amplify DDoS attacks against name servers. DNSSEC signs all records before they are used hence are vulnerable to replay attacks but DNSCurve signs records as and when they are needed and keeps its private keys online (Edge, 2009).

DNSSEC is more popular than DNSCurve because the latter only fixes the Dan Kaminsky cache poisoning attack explained in Section 2.4 and does not address other DNS security attacks such as MITM and DNS query redirection. It is similar to source port randomization in that they provide a solution to the same problem (Vixie, 2010). In addition, the net improvement on authoritative name server transport from using Curve25519 is just one and a half times better than DNSSEC (Kaminsky, 2011). DNSSEC also offers better protection against attacks where the attacker gets control of the server in that the private keys are kept off line as opposed to DNSCurve where the keys are kept on online (Kaminsky, 2011).

Caching is a desirable functionality of name servers and it helps to improve query turnaround and greatly reduce authoritative name server load. DNSCurve avoids usage of local caches in order to ensure security (Kaminsky, 2011).

3.9.2 TSIG

Transaction Signatures (TSIG) is defined as a Message Authentication Code (MAC) method that uses HMAC-MD5 to provide integrity and authentication of DNS messages between two communicating hosts in a computationally efficient manner (Andersson and Montag, 2008). The method uses a single key shared by the two communicating hosts to establish trust between themselves. RFC 2845 (Vixie *et al.*, 2000) highlights that TSIG can be used for secure zone transfers, dynamic updates and provision of simple and efficient secure communication between clients and servers in a single network. It cannot provide authentication and integrity for many hosts.

It is also documented as inappropriate for use between many servers in different networks as it will require each communicating pair to have a different set of signatures thereby requiring a large number of secret keys to be distributed (Vixie *et al.*, 2000). TSIG does not provide a method of exchanging the shared secret keys and systems administrators have to rely on other means like fax and mail to distribute the encryption key. The keys also require to be adequately secured and it is suggested that the keys should be changed once every 30-60 days.

TSIG is widely used to provide security to other parts of the DNS system like zone transfers and dynamic updates but it cannot cater for the entire domain name system because it lacks scalability (Kolkman, 2009). However, RFC 3130 (Lewis, 2001) explicitly defines TSIG as an integral part of DNSSEC because it provides authenticity and integrity to the DNS.

3.9.3 Administrative Security

It is essential that name server administrators adhere to strict security measures on their servers to minimize security risks (Aitchison, 2005). He explained that DNS systems will remain exposed to security vulnerabilities if basic administrative security measures are not implemented even if the zone is signed. The signing of a zone will increase security if server operating system, DNS software and permissions are securely configured.

Software patches are at times meant to upgrade known security vulnerabilities and in such instances upgrading should be done promptly. It is also advised to subscribe to advisory services that provide technology alerts on Bind and related DNS server software. Upgrading operating systems periodically are mentioned as good administrative security. However upgrading software for new features can wait while adequate tests are carried out and when finally done, detailed upgrade checklist should always be kept.

It is a good security measure to enable only necessary system features and disable all other features that are not needed. Defensive configuration where major security related features are explicitly enabled or disabled will help to do away with default settings which can be a security risk. Denying everything and allowing desired features selectively is also a good administrative security measure to limit functionality.

Limiting permission will enhance confidentiality by limiting access to files used by the DNS software and also prevents the DNS system from writing to other locations if compromised. DNS attacks like cache poisoning can be avoided if DNS system files have the right permissions. Seven Bind files that need adequate read/write protection are identified as named.conf, included files, zone files, pid files, log files, rndc files and journal files. Setting the correct permissions on these files will result in good security settings for Bind.

The author suggests running Bind on multiple operating system as a way of reducing a exposure to a single weakness. Different operating systems will most likely not share the same weaknesses thereby improving security. Running a different version of DNS software for mission-critical systems will be important in case the main system gets compromised.

3.10 DNSSEC Mitigation Against DNS Security Threats

Chapter 2 identified and explained the security threats that affect the DNS. This Section shows why organizations should deploy DNSSEC by bringing out how the security extensions to the DNS protocol address the security threats. It shows that DNSSEC mitigates all the DNS security threats save for the DoS attacks.

All the DNS security threats outlined in Chapter 2 take advantage of the fact that the DNS servers have limited ways of validating the source of the responses and trick them into accepting response from malicious sources. Brute force attacks are also used in the Kaminsky cache poisoning and birthday attacks. DNSSEC mitigates these security risks by providing security aware resolvers with a way of validating authenticity and integrity of responses they receive signed authoritative name servers.

A security aware name server sends both signed (RRSIG) and unsigned resource records to DNS queries from security aware resolvers. The signed resource records are decrypted using the authoritative name server's public keys and compared with the unsigned resource records. Any differences between the plain text resource records and the decrypted records will result in the dropping of the response for failing the validation process. Only response with matching records will be accepted as shown in Figure 3.8 below.

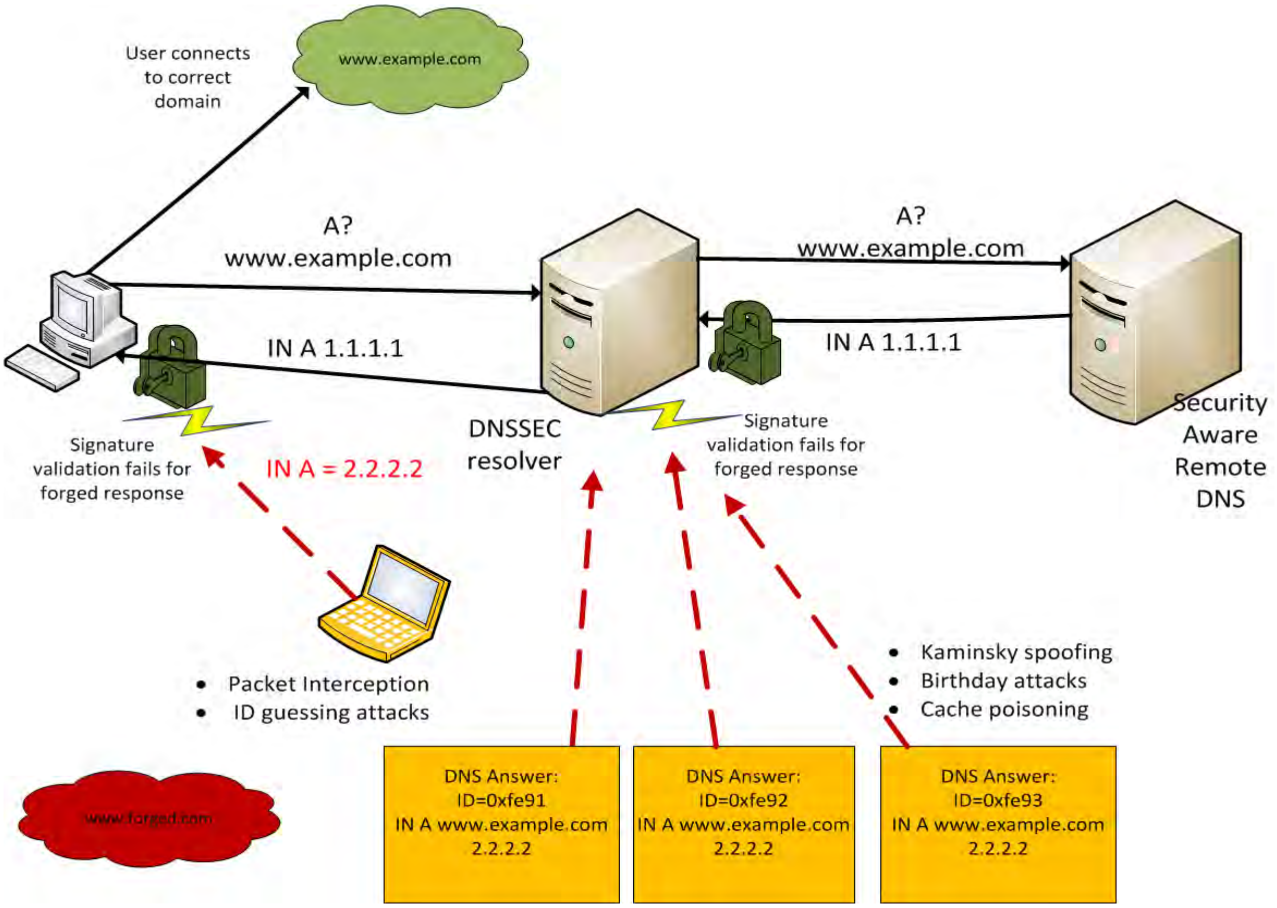

User connects to correct domain
www.example.com
A? www.example.com
A? www.example.com
IN A 1.1.1.1
IN A 1.1.1.1
Signature validation fails for forged response
IN A = 2.2.2.2
DNSSEC resolver
Signature validation fails for forged response
Security Aware Remote DNS
• Packet Interception
• ID guessing attacks
• Kaminsky spoofing
• Birthday attacks
• Cache poisoning
www.forged.com
DNS Answer: ID=0xfe91 IN A www.example.com 2.2.2.2
DNS Answer: ID=0xfe92 IN A www.example.com 2.2.2.2
DNS Answer: ID=0xfe93 IN A www.example.com 2.2.2.2

Figure 3.10: DNSSEC Security

All forms of cache poisoning including birthday attacks, Kaminsky cache poisoning will fail because their responses will fail signature validation process described above. ID guessing and query prediction attacks will also fail because of the absence of the correct digital signature.

The diagram shows that a stub resolver can be subjected to packet interception or MITM attacks if it does not have a local validating resolver. It is therefore important to secure the communication channel between the resolver and the client machine. For internet users relying on remote DNS servers provided by ISP's, use of a local validating resolver can prevent the security threats of the last mile. DNSSEC validating applications can also achieve the same objective but if they are available.

3.11 Summary

This chapter explored the structure of DNSSEC and the administrative tasks associated with managing a signed zone. The trends in the deployment of DNSSEC were also explored with special emphasis on Africa before looking at the future uses of DNSSEC. An evaluation of the benefits of signing a zone for TLD operators and organizations in general followed after which the methods of deploying DNSSEC for Internet and intranet users were analyzed. The Chapter concluded with the analysis of alternative security techniques for DNS and how DNSSEC mitigates DNS security threats mentioned in Chapter 2.

Chapter 4

Implementation

The previous chapter illustrated how DNSSEC can be used to mitigate against several known DNS security threats. It also shows that the majority of TLD's have not yet deployed DNSSEC. Zones that fall under TLD's that are not signed need to consider other options of deploying DNSSEC so that DNS security can be improved. Section 4 describes the steps to be carried out in order to evaluate the effectiveness of DLV, DNSSEC validating resolvers and web based validators. The Section also describe steps taken to determine the effect of DNSSEC on message sizes.

4.1 Purpose of the Study

This thesis aims at promoting the deployment of DNSSEC on a wider scale as a way of improving DNS security where the TLD is not signed. The primary objective is to find out how users out there, can make effective use of DNSSEC to ensure secure domain name to address mappings. The findings of this research can be used to help provide advice and guidelines on how users can have an additional layer of Internet security by using DNSSEC. In addition, the research seeks to find the effect of DNSSEC on DNS message sizes.

4.2 Methods of Deploying DNSSEC

When the TLD is signed, deploying DNSSEC involves the process of signing a zone and publishing trust anchors with the parent zone. This allows validation of the zone's trust anchors through the normal trust chain starting from the root down to the target domain. When the TLD is not signed, the chain of trust is broken and the child zone cannot publish their keys with an unsigned parent. The following methods can be used to deploy DNSSEC in such islands of security.

1. **Domain Look-aside Validation (DLV) domains** - DLV is a server side mechanism that allows a zone in island of security to publish their trust anchors in the DLV domain. Zones also use the DLV domain to validate the trust anchors of other domains. Since only two TLD's are signed in Africa, this method provides a viable alternative for deploying DNSSEC for many organizations across Africa. Currently, there are two trust anchor repositories namely ISC[1] and Secspider[2], that can be used as DLV domains.

2. **Local Resolvers** - Local resolvers are recursive name servers installed on a client computer to validate DNSSEC responses from security aware name servers as shown in figure 4.1. This definition does not include stub resolvers that cannot perform DNSSEC validation. The validating resolver at A in Figure 4.1 validates responses at the client levels. They advertise that they are security aware by including the DNSSEC OK bit when making queries. Subsequently, they receive DNS responses with security data and validate them for integrity and authenticity. This solution can be used by both Internet and intranet users for improved DNS security on a client computer level. Client PC B does not have a local resolver and get their responses from the ISP or corporate DNS servers. Client PC B does not have a local resolver and get their responses from the ISP or corporate DNS servers.

3. **Web based validators** - Web browser validators are a client side method that brings visibility to DNSSEC validation. They present the DNSSEC status of the domain their are accessing by using different colors for the DNSSEC key icon in the web browsers. As illustrated in figure 4.2, green color in the icon indicates correct DNSSEC validation, while red indicates failure. They can be configured to use resolvers of the user's choice. When a domain's DNSSEC status cannot be verified, the key icon turns grey. Currently, web browser validators are available for Google Chrome, Internet Explorer and Mozilla Firefox.

[1] https://dlv.isc.org/

[2] http://secspider.cs.ucla.edu/

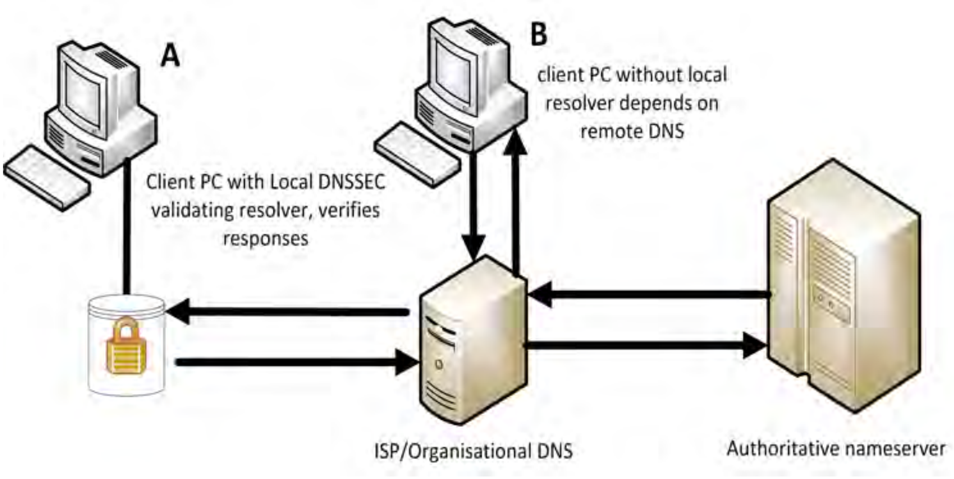

A
Client PC with Local DNSSEC validating resolver, verifies responses
B
client PC without local resolver depends on remote DNS
ISP/Organisational DNS
Authoritative nameserver

Figure 4.1: Using validating resolvers

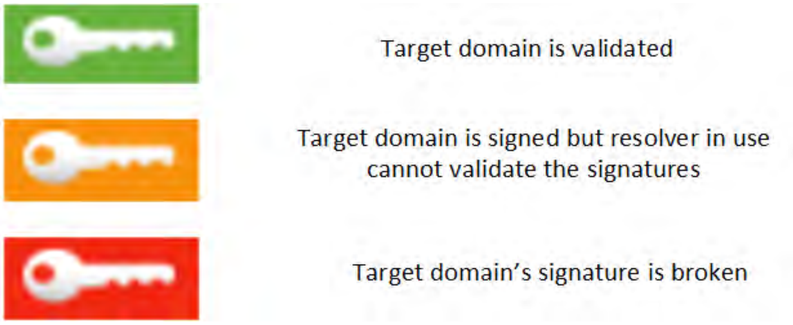

Target domain is validated
Target domain is signed but resolver in use cannot validate the signatures
Target domain's signature is broken

Figure 4.2: Web based validator icons

4.3 Experimental Design

In order to evaluate the effectiveness of DLV, a zone will be created in an unsigned TLD. The zone will be signed and registered with a trust anchor repository. The resource records in the signed zone will be queried from security aware and security oblivious client computers to determine the effectiveness of using DLV domains for deploying DNSSEC. The zone will be further tested and analyzed using online tools such as Secspider[3], DNSViz[4] and DNSSEC Analyzer[5].

For testing resolvers, DNSSEC validating resolvers will be installed on Windows and Linux client computers. Each in order to test local resolvers. Web browser validators will be tested on the same client computers. The software, key management practices that will be used in the tests are detailed below.

[3] http://secspider.cs.ucla.edu/
[4] http://dnsviz.net/
[5] http://dnssec-debugger.verisignlabs.com/

1. **Zone**

   In order to minimize the research costs, an existing and unsigned domain, *moria.org* will be used in the test. A child domain to moria.org will be created. It will be signed and be registered with a DLV domain. The zone will have a small number of resource records that are only sufficient for the tests and will therefore be hosted on the same name server with its parent. The zone will be given the name *demo.moria.org.*

2. **Software**

   Bind 9.8, the stable version of the most commonly used DNS software will be used for the demo.moria.org name servers. Bind 9.8 comes with automatic zone testing tools such as *donuts* and *named-checkconf.* It also ships with automatic key management and rollover tools that include *zonesigner*[6] and *rollerd*[7]. Zonesigner is an open source tool for signing a zone on the command line. Rollerd is a command line tool that is used for key rollover. Windows 7 and Ubuntu 12.04 will be used to test local resolvers and web browser validators. Unbound[8] and DNSSEC Trigger[9] are the DNSSEC enabled local resolvers that will be used for resolver tests.

3. **Key Management**

   Separate KSK and ZSK for signing the test zone as recommended in ENISA (Lowinder *et al.*, 2010) and NIST (Chandramouli and Rose, 2010) publications. Recommended key lengths recommendations of 1024 and 2048 for the ZSK and KSK respectively will be used in the *demo.moria.org* zone. An initial key lifetime of 30 days will be used to afford the research to experience key rollover. Longer key lifetime will be used thereafter to allow the research to focus on the research objectives.

4.4 Testing

This Section explains in detail the exact tests that will be done in the deployment of DNSSEC in environments where the TLD is not signed. The three methods that can be used to deploy DNSSEC will be tested individually to determine their suitability for use by the two groups of users identified in Section 3.5. The fourth test will focus on the increase in DNS message size when a zone is signed. The results of this test will be used

[6] *https://www.dnssec-tools.org/wiki/index.php/Zonesigner*

[7] https://www.dnssec-tools.org/wiki/index.php/Rollerd

[8] http://unbound.net/

[9] http://www.nlnetlabs.nl/projects/dnssec-trigger/

to determine whether it is necessary for an average organization to acquire additional hardware when deploying DNSSEC.

4.4.1 DNSSEC Look-Aside Validation (DLV)

The review of DNSSEC in chapter 3 highlighted that the root domain has been entirely signed but several other TLD remain unsigned. Many countries TLD's remain unsigned without any attempts to sign them at present. A lot of these countries are in Africa where only 2 countries on the continent have signed their TLDs. Organizations in these countries that use the ccTLD will live without a signed parent domain for some time hence it is necessary to employ DLV domains to publish their zone trust anchors.

DLV presents a way for a signed domain in an island of security to receive and provide authenticated responses as specified in RFC 5074 (Weiler, 2007). In instances where the TLD is signed, DLV can be used to test a signed zone before going live (Aitchison, 2005). Governments can also implement DLV as disaster recovery technique for a ccTLD in the event that the root domain is down for some reasons like random failure. DLV can also be used to enhance security for organizations that offer security sensitive services like financial institutions and online traders.

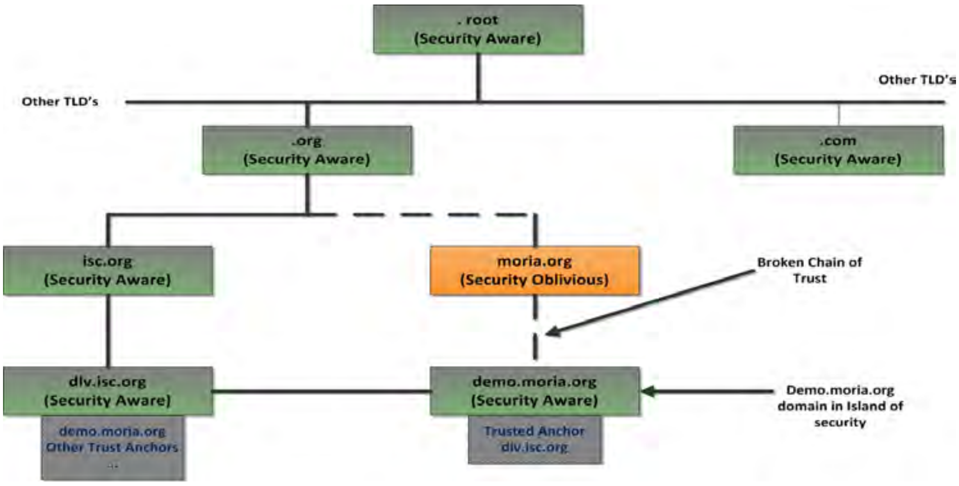


Figure 4.3: DLV test

The demo.moria.org zone will be signed and its trust anchors distributed to Internet Systems Consortium (ISC) DLV domain. The ISC trust anchor repository was chosen on the strength of its clearly spelt DLV policy (ISC, 2009a). It also provides an easy way of uploading trust anchors. In addition, it also performs checks to determine zone administrative rights of the person submitting the zone trust anchors. With their lengthy involvement with DNSSEC, ISC is expected to offer better DLV services.

The zone demo.moria.org will be configured as shown in Figure 4.3. This will allow publication of trust anchors in isc.dlv.org. Security aware resolvers should be able to obtain responses with security data from the test zone using the DLV facility.

Test Procedure

1. Configure domain to use DLV.
2. Sign the domain specifying the ISC DLV.
3. Submit domain trust anchors to the ISC DLV domain.
4. Test for correct DLV configuration.
5. Test provision of security data to DNS queries.

Expected Results

1. Secure domain through ISC DLV link.
2. Provision of security data to DNS queries.
3. The **AD** flag should be present in responses provided to other security aware name servers.

Data Presentation

The data on successful and failed DNSSEC queries in each scenario will be presented in a tabular format.

4.4.2 DNSSEC Validating Resolvers

Installing local validating resolvers is a solution that can be used by both intranet and Internet users. The local resolvers will do security validation on the client PC as opposed to name server validation. It removes the need to secure the communication channel between the name server and the client computer. This solution can be implemented in all the instances of the DNSSEC matrix presented in chapter 3 but it will be more useful for Internet users whose ISP operates an unsigned domain.

Two applications that can be used for this purpose are DNSSEC Trigger[10] and Unbound[11].

[10] http://www.nlnetlabs.nl/projects/dnssec-trigger/

[11] http://unbound.net/

Both of them work on both Windows and Linux are perfect for our tests on Windows 7 and Ubuntu 12 Client machines. The local validating resolvers will be tested before the domain has been signed with much emphasis on investigating if they can do security validation when the domain is not signed.

Test Procedure

The steps listed below provide a precise description of how this test will be accomplished.

1. Install DNSSEC Trigger on Windows 7 and Unbound on Ubuntu 12.04 client.
2. Send queries to both signed and unsigned domains and observe results using dig command and web browsers.
3. Send DNS queries to domains with deliberately misconfigured zones to verify if the validators can detect the misconfiguration.
4. Observe and record test results.

Expected Results

1. The root key file updated upon connection to the Internet.
2. AD bit in responses from signed domains.
3. No AD bit in all responses from unsigned domains.
4. Authentication failure for mis-configured domains.

Data Presentation

A graphical representation of data showing comparing DNSSEC validations for the two resolvers.

4.4.3 Web Based Validators

The DNSSEC literature review noted that end user applications that can do DNSSEC validation will provide better security to Internet users as opposed to using security aware name servers. We only identified the web browser plug-ins (Dnssec Deployment Initiative, 2012) as the available DNSSEC enabled applications. The web browser plug-ins for

Mozilla Firefox, Google Chrome and Internet Explorer will be tested for their ability to accurately validate DNS responses. The Internet is one of the most critical use of the DNS hence it is important that it has tools to inform users about the security status of the domain they are visiting. All the web browsers can be used on computers running on Windows but Internet Explorer is not available for Linux platforms.

The web browser validators can be very useful for Internet functions that require security like Internet banking and e-commerce. This can also motivate providers of these security services to deploy DNSSEC. The three web browsers will be installed on Windows 7 and Ubuntu 12.04 client computers. Tests will be done through visiting web addresses from signed, unsigned and deliberately misconfigured domains (DNS-OARC, 2012a). Further tests will be conducted to find out whether using remote and local validating resolvers separately gives the same results. Visits to non-secured sites to find out if the web browsers can provide reliable results will be done. Detailed steps for testing the web browser validators are listed below.

Test Procedure

1. Install DNSSEC web browser plug-ins on Windows 7 and Ubuntu 12.04 ins on client computers.
2. Test the validators using local DNSSEC aware, local DNSSEC oblivious and remote DNSSEC aware name servers
3. Visit secured sites and record results.
4. Visit non-secured sites and record results.
5. Visit mis-configured sites and record results.
6. Summarize the results and interpret the results.

Expected Results

The validators should have a 100% rate of accuracy for them to be depended on for providing security for clients without security aware name servers.

Data Presentation

The experiment results will be presented in a tables analyzing the behavior of the three browsers plug-ins in the three test cases outlined above.

4.4.4 Message Size

It has been reported that DNSSEC results in huge DNS queries. EDNS0 (Vixie, 1999) extends DNS packet size from the 512 bytes limit stated in RFC 1035 (Mockapetris, 1987a) to 4096 bytes. This test procedure's objective is to find the extent to which DNSSEC increases DNS message sizes. The test will also analyze usage patterns for both UDP and TCP for DNSSEC queries.

Wireshark[12] , an open source network traffic analyzer tool, will be used to measure message sizes for both security oblivious and security aware resolvers. Wireshark provides for live capture of computer network traffic and off line analysis of the data. In addition, it can be used across multiple platforms that include Ubuntu and Windows. Further, it allows saving of network data files in many different formats and has a user friendly graphical interface as opposed to similar tools.

Test Procedure

The first tests will be done on non-DNSSEC resolvers. The second batch of tests will be conducted after the test zone has been signed and DLV configured. The steps listed below will complete this test:

1. Install Wireshark and let it record DNS queries.
2. Send DNS queries before signing the zone using web browsers.
3. Save the query results.
4. Sign the zone and configure the resolver to accept only DNSSEC responses.
5. Send queries to signed domains using web browsers.
6. Save the results to a separate file.
7. Compare and analyze message sizes for the 2 test cases.

Expected Results

1. Messages with security data will be bigger than messages without security data.
2. Some DNSSEC messages will use the TCP instead of UDP.

[12] http://www.wireshark.org/

Data Presentation

Data collected will be presented using line graphs to give a comparison of message sizes for DNSSEC and non-DNSSEC messages.

4.5 Summary

The first Section of this chapter looks back at the purpose of research before discussions on the technical setup of the experimental design are given in the second section. Highlights of the tests that are going to be conducted on DLV, local resolvers and web browser validators in order to meet the objectives of this research are also given. The results collected during the tests will be discussed in the next chapter. The tests focused on methods for deploying DNSSEC in islands of security as this is the problem area that this research attempts to address.

Chapter 5

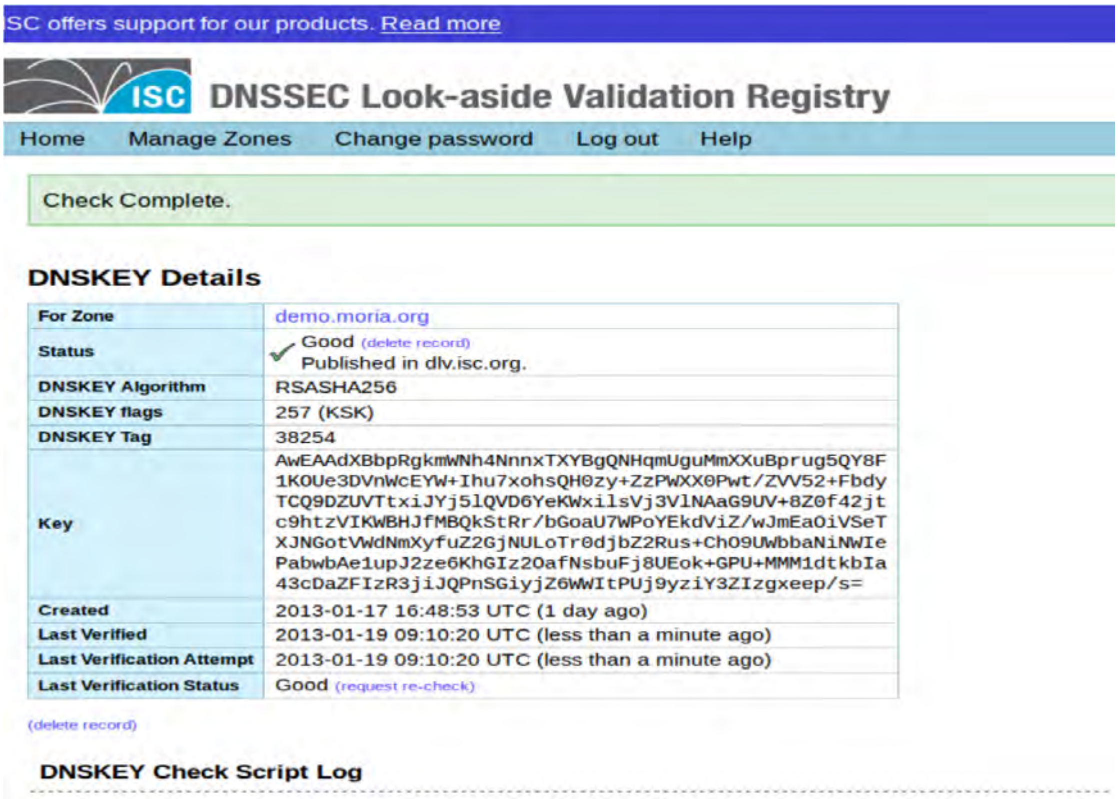
Results

This chapter analyses the results of the tests done on using DLV, validating resolvers and web validators for improving DNS security. Section 5.1 provides an analysis of the results obtained from testing DLV domains for DNSSEC validation whilst Sections 5.2 and 5.3 present results for validating resolvers and web browser validators respectively. These three methods can be employed by both Internet and intranet clients to use DNSSEC when their parent domains are not signed. Analysis of the results and recommendations on how best to make use each method is done after presentation of the results.

5.1 Domain Look-Aside Validation

The domain demo.moria.org was created and signed using ISC DLV trust anchor repository. The first step is creating a ISC DLV registry profile which is used for uploading the trust anchors and other zone management processes. After creating the profile, uploading of DS and DNSKEY resource records through a web interface follows. There is a requirement to add a TXT resource record in the domain to prove possession of domain administration rights. The TXT resource record was added and the zone resigned. Proving zone administrative rights through adding the TXT resource record is done every time the KSK are changed.

Figure 5.1 shows the ISC DLV registry web interface for managing zone keys after successful upload of the demo.moria.org DNSKEY. It also shows the DNSKEY uploaded, its key tag and algorithm.


SC offers support for our products. Read more
ISC
DNSSEC Look-aside Validation Registry
Home
Manage Zones
Change password
Log out
Help

Check Complete.

DNSKEY Details

For Zone	demo.moria.org
Status	✓ Good (delete record) Published in dlv.isc.org.
DNSKEY Algorithm	RSASHA256
DNSKEY flags	257 (KSK)
DNSKEY Tag	38254
Key	AwEAAdXBbpRgkmWNh4NnnxTXYBgQNHqmUguMmXXuBprug5QY8F 1KOUe3DVnWcEYW+Ihu7xohsQH0zy+ZzPWXX0Pwt/ZVV52+Fbdy TCQ9DZUVTtxiJYj5lQVD6YeKWxilsVj3VlNAaG9UV+8Z0f42jt c9htzVIKWBHJfMBQkStRr/bGoaU7WPoYEkdViZ/wJmEaOiVSeT XJNGotVWdNmXyfuZ2GjNULoTr0djbZ2Rus+ChO9UWbbaNiNWIe PabwbAe1upJ2ze6KhGIz2OafNsbuFj8UEok+GPU+MMM1dtkbIa 43cDaZFIzR3jiJQPnSGiyjZ6WWItPUj9yziY3ZIzgxeep/s=
Created	2013-01-17 16:48:53 UTC (1 day ago)
Last Verified	2013-01-19 09:10:20 UTC (less than a minute ago)
Last Verification Attempt	2013-01-19 09:10:20 UTC (less than a minute ago)
Last Verification Status	Good (request re-check)

(delete record)

DNSKEY Check Script Log

Figure 5.1: ISC DLV Web interface

5.1.1 Testing DLV Configuration

Web based tools used to analyze signed domain names for correct DNSSEC configuration were used to verify correct configuration of demo.moria.org. DNSSEC Analyzer and DNSViz were the two tools used for this test. DNSSEC Analyzer checks for problems along the authentication chain and lists all the DS and DNSKEY resource records that define the authentication chain from the root to the target zone. The sub zone demo.moria.org is listed as insecure as there are no DS or DNSKEY records in the parent moria.org domain. This is because the tool does not recognize the existence of a DLV domain. However the test shows the zone contains DNKEY RR and other RRISGs which means it is signed.

DNSViz presents a detailed analysis of the zone that includes the DLV repository. The presentation in Figure 5.2 was taken from the web analysis results of demo.moria.org. The digram shows a broken chain of trust from the root to demo.moria.org. The domain moria.org is not signed and its child domain demo.moria.org has to upload its trust anchors at isc.dlv.org. The diagram shows that to form a chain of trust the zone demo.moria.org is linked to the root zone through dlv.isc.org. The parent zone is not signed and cannot be used for DNSSEC hence demo.moria.org distributes its trust anchor. The panel on the left also shows that there is a secure delegation from dlv.isc.org to demo.moria.org. In addition, this tool also shows other zone information like DNSKEY RR and their respective key id's.

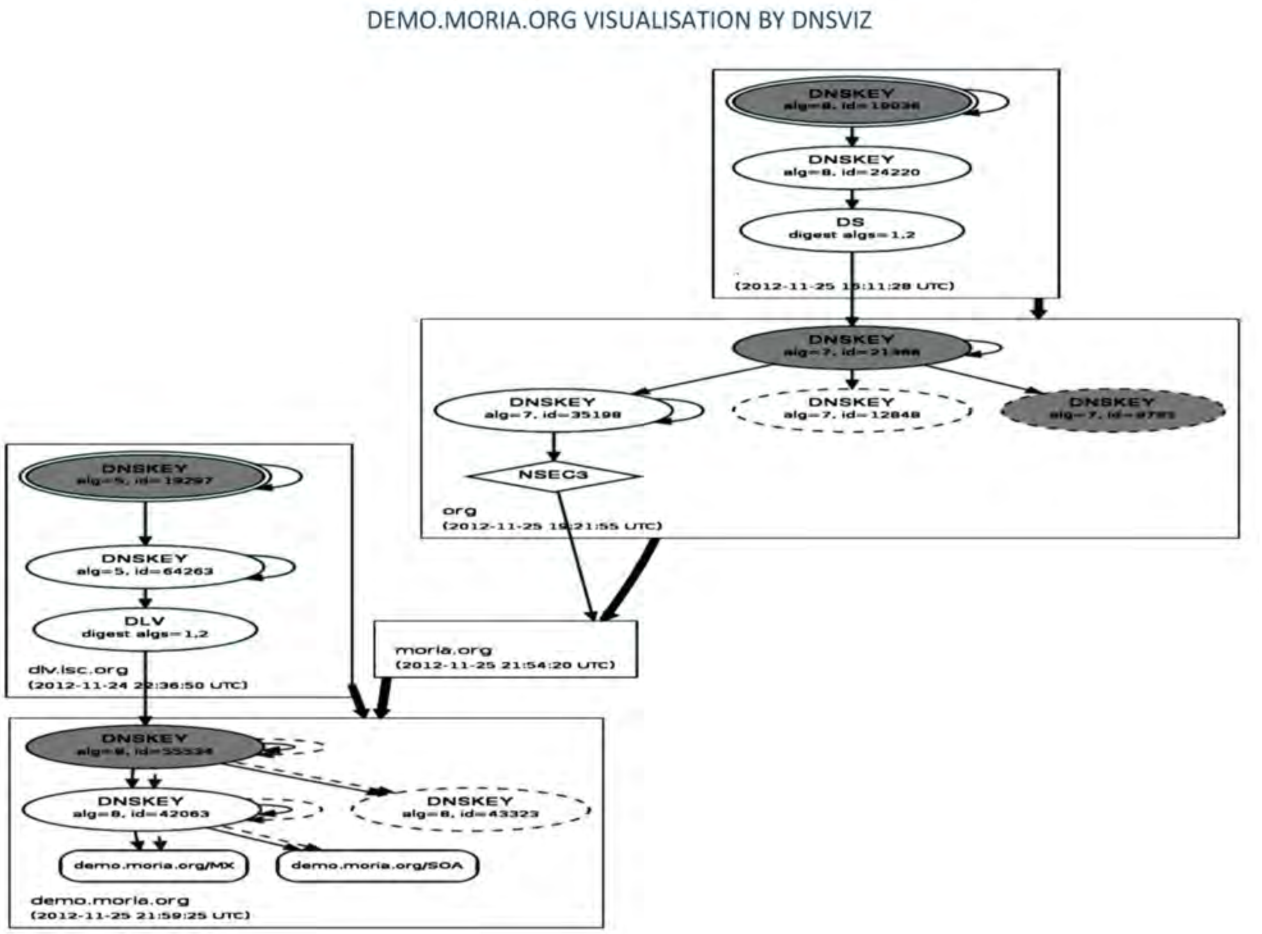
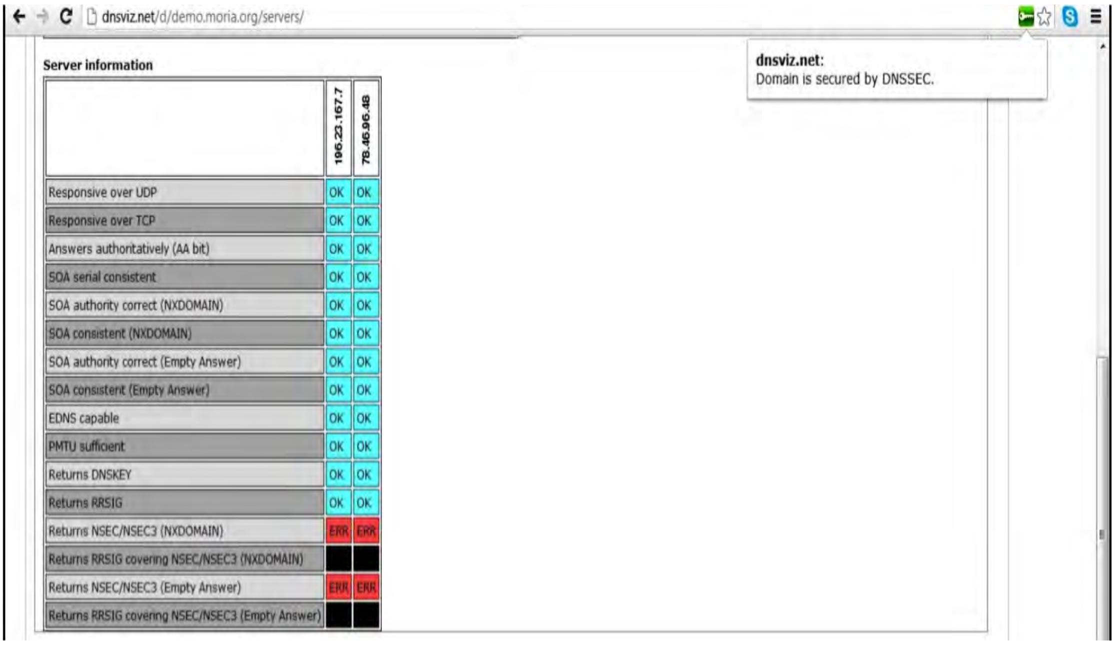


Figure 5.2: DNSViz Zone Visualization (DNSViz, 2013a)

DNSViz also provide other useful information like the status of the DNS servers that host the signed zone. This information includes the server name, parent and child zone information. More importantly this tool also has a section entitled server information which contains important server properties that can affect DNSSEC. These properties include whether the servers can use both UDP and TCP, server's EDNS0 capabilities and they return DNSKEY and RRSIG. The information for the servers that host demo.moria.org is shown in Figure 5.3. The server information is very critical for DNSSEC troubleshooting. For example, both primary and secondary servers used by demo.moria.org do not support NSEC3 as shown in Figure 5.3



dnsviz.net:
Domain is secured by DNSSEC.

Server information

	196.23.167.7	78.46.96.48
Responsive over UDP	OK	OK
Responsive over TCP	OK	OK
Answers authoritatively (AA bit)	OK	OK
SOA serial consistent	OK	OK
SOA authority correct (NXDOMAIN)	OK	OK
SOA consistent (NXDOMAIN)	OK	OK
SOA authority correct (Empty Answer)	OK	OK
SOA consistent (Empty Answer)	OK	OK
EDNS capable	OK	OK
PMTU sufficient	OK	OK
Returns DNSKEY	OK	OK
Returns RRSIG	OK	OK
Returns NSEC/NSEC3 (NXDOMAIN)	ERR	ERR
Returns RRSIG covering NSEC/NSEC3 (NXDOMAIN)	■	■
Returns NSEC/NSEC3 (Empty Answer)	ERR	ERR
Returns RRSIG covering NSEC/NSEC3 (Empty Answer)	■	■

Figure 5.3: DNSViz Server Information (DNSViz, 2013b)

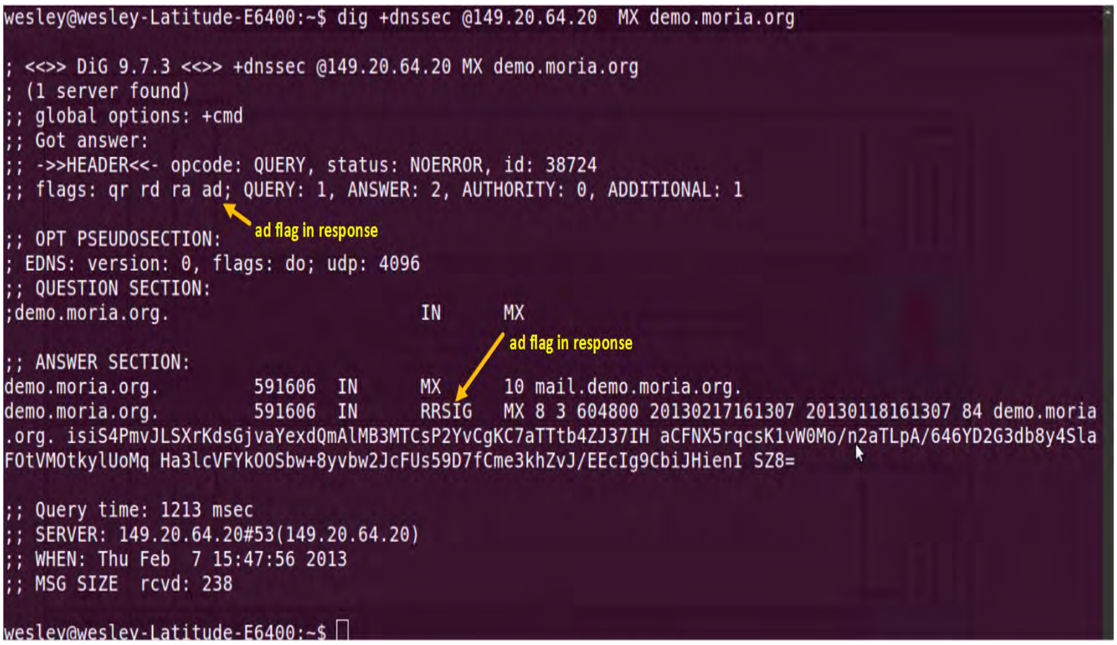
5.1.2 Querying demo.moria.org RRs

After verifying correct zone signing and DLV configuration, another critical test is to find out if the zone can provide security data for queries on RRs in its zone. This test was accomplished by using the ***dig*+dnssec** command. The command can be used with either a local or remote DNSSEC validating resolver. The earlier option was used because the zone demo.moria.org was configured as authoritative only name server. The signed DNS-OARC[1] remote resolver was used to query for demo.moria.org resource records.

Queries were made for resources records that exist in the demo.moria.org zone: MX, NS, SOA, DNSKEY and A. Responses for all these queries had the authenticated data flag and each resource record was accompanied by a corresponding RRSIG as illustrated in Figure 5.4. In Figure 5.4, a query is made for the MX resource record from demo.moria.org. The response obtained has an ***ad*** flag and the MX RRSIG which is used to authenticate the unencrypted MX RR.

The full results of the queries for resource records in demo.moria.org are shown in Table 5.1. All the responses obtained using the ***dig +dnssec*** command had the **ad** flag and RRSIGs. The results concur with DNSViz responses shown in results obtained from DNSViz web based tests under the responses Section (DNSViz, 2013c).

[1] https://www.dns-oarc.net/oarc/services/odvr



```
wesley@wesley-Latitude-E6400:~$ dig +dnssec @149.20.64.20  MX demo.moria.org

; <<>> DiG 9.7.3 <<>> +dnssec @149.20.64.20 MX demo.moria.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38724
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;demo.moria.org.                        IN      MX

;; ANSWER SECTION:
demo.moria.org.         591606  IN      MX      10 mail.demo.moria.org.
demo.moria.org.         591606  IN      RRSIG   MX 8 3 604800 20130217161307 20130118161307 84 demo.moria
.org. isiS4PmvJLSXrKdsGjvaYexdQmAlMB3MTCsP2YvCgKC7aTTtb4ZJ37IH aCFNX5rqcsK1vW0Mo/n2aTLpA/646YD2G3db8y4Sla
FOtVMOtkylUoMq Ha3lcVFYkOOSbw+8yvbw2JcFUs59D7fCme3khZvJ/EEcIg9CbiJHienI SZ8=

;; Query time: 1213 msec
;; SERVER: 149.20.64.20#53(149.20.64.20)
;; WHEN: Thu Feb  7 15:47:56 2013
;; MSG SIZE  rcvd: 238

wesley@wesley-Latitude-E6400:~$
```

Figure 5.4: dig test

Table 5.1: Demo.moria.org RR query results

Resource Record	AD Flag	RRSIG
SOA	yes	yes
NS	yes	yes
MX	yes	yes
DNSKEY	yes	yes
A	yes	yes

5.2 DNSSEC Validating Resolvers

Validating resolvers installed on client machines include the DNSSEC OK bit in their queries to indicate that they are able to process signed responses. When they query signed name servers they will receive responses with security data. The security data is verified using the chain of trust starting with the root trust anchors configured within the resolver. Users can decide whether they want to receive only validated answers only or both validated and unvalidated through appropriate configurations of local resolvers.

Three pools of signed, unsigned and deliberately misconfigured domains were used to test validating resolvers. Two validating resolvers, namely Unbound and DNSSEC Trigger were used. Tables 5.2, 5.3 and 5.4 show the domains used to conduct this test.

Table 5.2: Signed Domains

Domain Name	**DNSSEC Status**
isc.org	signed
internetsociety.org	signed
dnsviz.net	signed

Table 5.3: Unsigned Domains

Domain Name	**DNSSEC Status**
google.com	unsigned
yahoo.com	unsigned
dnssec.net	unsigned

Table 5.4: Deliberately misconfigured domains

Domain Name	**DNSSEC Status**
rhybar.cz	misconfigured
dnssec-failed.org	misconfigured
forged.test.xelerance	misconfigured

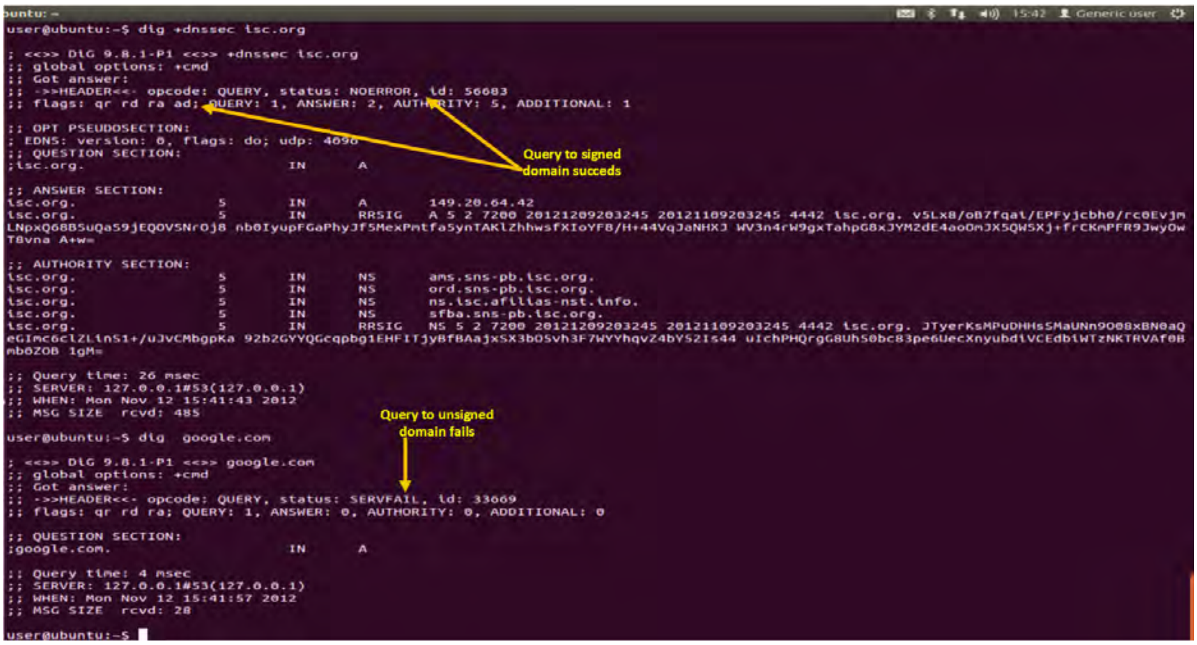
5.2.1 Unbound

Unbound 1.46 was configured on Ubuntu 12.04 client for the purpose of assessing the effectiveness of a local resolver in validating DNSSEC responses. The overall objective was to determine whether a user can depend on Unbound for security against DNS security threats. The first tests were done with the default Unbound configuration that drops all responses that fail DNSSEC validation. Whilst this configuration offers the greatest defense against DNS security threats, it may be inappropriate for use at this point in time when many organizations have not implemented DNSSEC. If the default configuration is used, all unsigned domains will not be accessible.

Changing the configuration to allow responses that fail validation enables users to access domains that are not yet signed. This may be necessary especially now when many zones are not signed.

The command ***dig +dnssec domain name*** was used to do the first rounds of tests. The tests from the pool of signed domains used in this test retained NOERROR responses. In addition, the query responses had the ***ad*** flag reflecting correct validation of the responses. Responses from unsigned domains and misconfigured domains retained SERVFAIL error message as shown in Figure 5.5.

The unbound configuration was changed to allow responses from unsigned domains by setting ***val-permissive-mode: yes*** in the Unbound configuration file. After the changes,



```
user@ubuntu:~$ dig +dnssec isc.org

; <<>> DiG 9.8.1-P1 <<>> +dnssec isc.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56683
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;isc.org.                       IN      A

;; ANSWER SECTION:
isc.org.                5       IN      A       149.20.64.42
isc.org.                5       IN      RRSIG   A 5 2 7200 20121209203245 20121109203245 4442 isc.org. v5Lx8/oB7fqal/EPFyjcbh0/rc0Evjm
LNpxQ68BSuQaS9jEQOVSNrOj8 nb0IyupFGaPhyJf5MexPmtfa5ynTAKlZhhwsfXIoYF8/H+44VqJaNHXJ WV3n4rW9gxTahpG8xJYM2dE4aoOmJX5QWSXj+frCKmPFR9JwyOw
T8vna A+w=

;; AUTHORITY SECTION:
isc.org.                5       IN      NS      ams.sns-pb.isc.org.
isc.org.                5       IN      NS      ord.sns-pb.isc.org.
isc.org.                5       IN      NS      ns.isc.afilias-nst.info.
isc.org.                5       IN      NS      sfba.sns-pb.isc.org.
isc.org.                5       IN      RRSIG   NS 5 2 7200 20121209203245 20121109203245 4442 isc.org. JTyerKsMPuDHHsSMaUNn9O08xBN0aQ
eGImc6clZLinS1+/uJvCMbgpKa 92b2GYYQGcqpbg1EHFITjyBfBAajxSX3bOSvh3F7WYYhqvZ4bYS2Is44 uIchPHQrgG8Uh50bc83pe6UecXnyubdiVCEdbiWTzNKTRVAf0B
mb0ZOB 1gM=

;; Query time: 26 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Nov 12 15:41:43 2012
;; MSG SIZE  rcvd: 485

user@ubuntu:~$ dig  google.com

; <<>> DiG 9.8.1-P1 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 33669
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      A

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Nov 12 15:41:57 2012
;; MSG SIZE  rcvd: 28

user@ubuntu:~$
```

Query to signed domain succeds

Query to unsigned domain fails

Figure 5.5: Unbound DNSSEC Tests

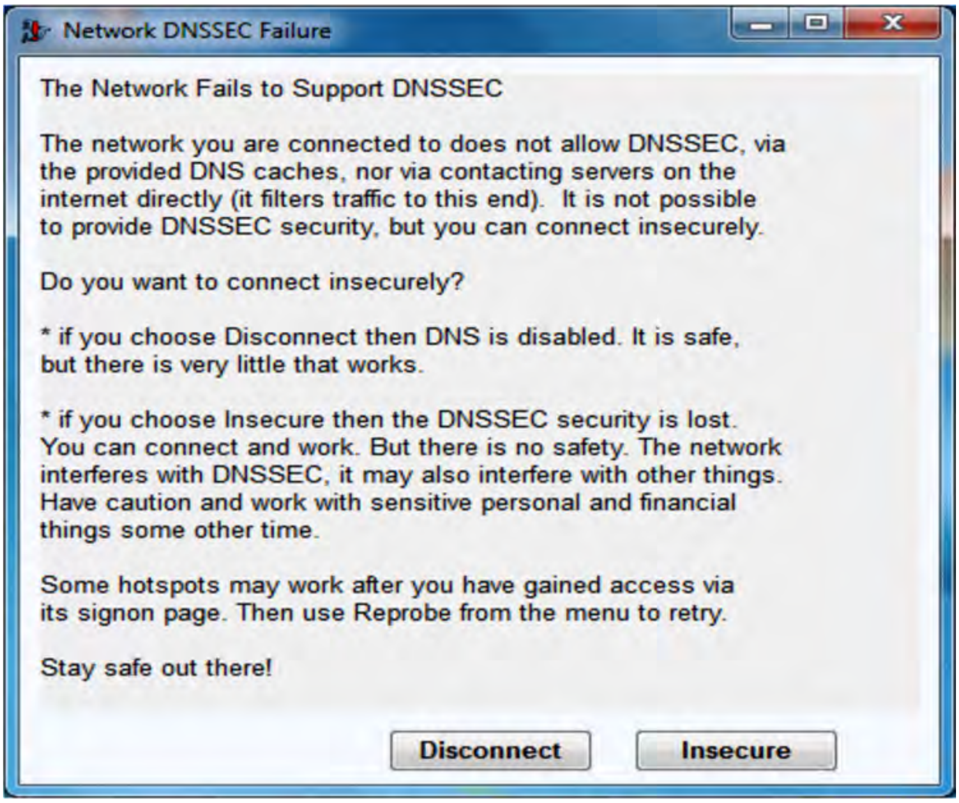
signed domains still replied with the ***ad*** flag and RRSIGs when queried with the ***dig +dnssec*** command. Unsigned domains had NOERROR responses but didn't supply security data as indicated by the absence of the ***ad*** flag and RRSIGs in the responses. Misconfigured domains retained SERVFAIL responses.

Further tests were conducted using web browsers installed on the Ubuntu client. Attempts to visit sites for signed, unsigned and misconfigured domains used in the first tests were made. Websites from signed domains were displayed while those from unsigned and misconfigured domains were not displayed. The error message displayed by Mozilla Firefox did not inform the user in any way that the unavailability was caused by DNSSEC validation failure.

5.2.2 DNSSEC Trigger

DNSSEC Trigger was installed on Windows 7 client to determine if it can correctly validate DNSSEC responses in a Windows environment. DNSSEC Trigger uses Unbound as the validating resolver but it comes with a daemon that probes the network to determine the best DNS forwarders to use in order to get DNSSEC. Unbound will perform DNSSEC validation but use the forwarders discovered by the DNSSEC Trigger daemon for query resolution.

When no suitable DNSSEC supporting name servers are found DNSSEC Trigger will notify the user with a message shown in Figure 5.6. The user can decide to connect insecurely or to disconnect. The disconnect option will stop the resolver from functioning and as a result all other domain names, signed or unsigned will not be reachable.



The Network Fails to Support DNSSEC

The network you are connected to does not allow DNSSEC, via the provided DNS caches, nor via contacting servers on the internet directly (it filters traffic to this end). It is not possible to provide DNSSEC security, but you can connect insecurely.

Do you want to connect insecurely?

* if you choose Disconnect then DNS is disabled. It is safe, but there is very little that works.

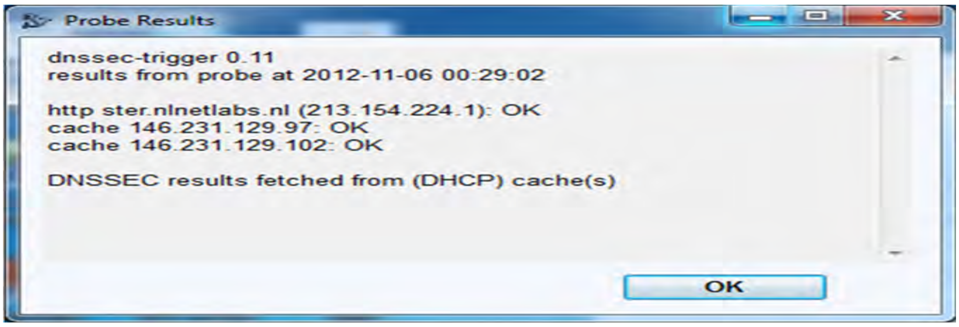
* if you choose Insecure then the DNSSEC security is lost. You can connect and work. But there is no safety. The network interferes with DNSSEC, it may also interfere with other things. Have caution and work with sensitive personal and financial things some other time.

Some hotspots may work after you have gained access via its signon page. Then use Reprobe from the menu to retry.

Stay safe out there!

Figure 5.6: DNSSEC Trigger Failure

DNSSEC Trigger daemon will continually probe the network for changes and will display the forwarders in use at an given time when the user checks the probe results. Figure 5.7 shows a typical probe result showing the name server that Unbound was using as query forwarders.



```
dnssec-trigger 0.11
results from probe at 2012-11-06 00:29:02

http ster.nlnetlabs.nl (213.154.224.1): OK
cache 146.231.129.97: OK
cache 146.231.129.102: OK

DNSSEC results fetched from (DHCP) cache(s)
```

Figure 5.7: DNSSEC Trigger: Probe Results

When using DNSSEC name servers DNSSEC Trigger correctly validates signed domains.

Web browser validators detected correct DNSSEC validation as shown DNSSEC Trigger test results for :signed, unsigned and misconfigured zones

5.2.3 Comparing Unbound and DNSSEC Trigger

The tests done show that Unbound is more effective as a local validating resolver than DNSSEC Trigger because DNSSEC Trigger sometimes fails to connect to remote resolvers to validate DNS queries as shown in Figure 5.8. DNSSEC Trigger however, communicates with the user when it fails and offers an option to go insecure. The graph below shows that Unbound correctly validated all queries sent to it. A total 81% of queries sent to DNSSEC Trigger were correctly validated. DNSSEC Trigger failed to validate 19% of the queries when it failed to connect to remote validating resolvers.

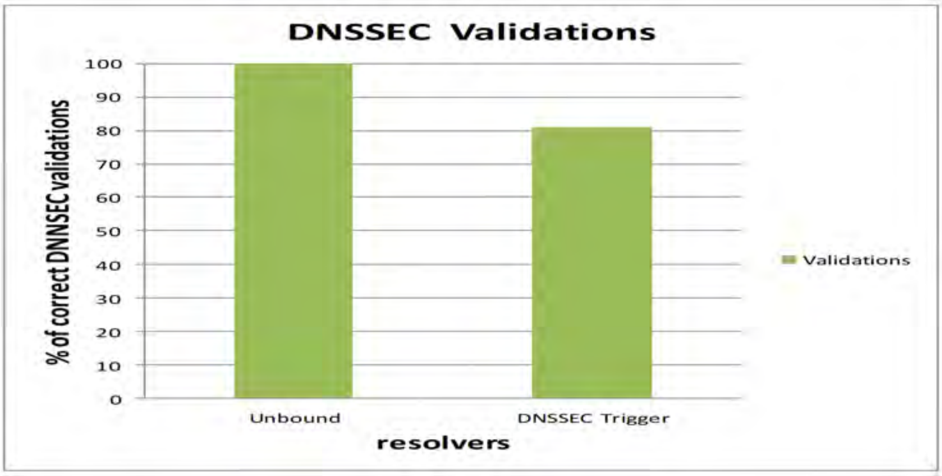


Figure 5.8: Comparing Unbound and DNSSEC Trigger

The failure to validate signed responses by client computer using DNSSEC Trigger is attributed to failure to obtain remote DNSSEC forwarding resolvers. On the occasions that DNSSEC validation failed, DNSSEC Trigger issued out an error message shown in Figure 5.6.

5.3 Browser Validators

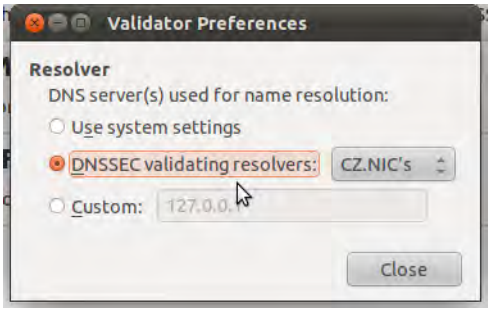
The DNSSEC validating plug-ins for Mozilla Firefox[2], Internet Explorer[3] and Google Chrome[4] were downloaded and installed. The plug-ins allows the user to chose DNS

[2] http://www.dnssec-validator.cz/

[3] http://www.dnssec-validator.cz/ie/

[4] http://labs.nic.cz/page/990/rozsireni-dnssec-validator-pro-google-chrome/

servers to use. Three DNS servers can be selected, one at a time. These are public DNSSEC validating resolvers provided by CZ.NIC and OARC, 127.0.01 and system supplied name servers. When 127.0.0.1 is selected, the local validating resolver will be used. System supplied name servers are those that are configured by default by the system settings. The public validating resolvers are remote open recursive DNS servers provided for public use by different organizations. System supplied settings refers to default name servers supplied by organizations and ISP's for intranet and Internet users respectively. Figure 5.9 shows the DNS server options in Mozilla Firefox.



Validator Preferences

Resolver

DNS server(s) used for name resolution:

○ Use system settings

● DNSSEC validating resolvers: CZ.NIC's

○ Custom: 127.0.0.1

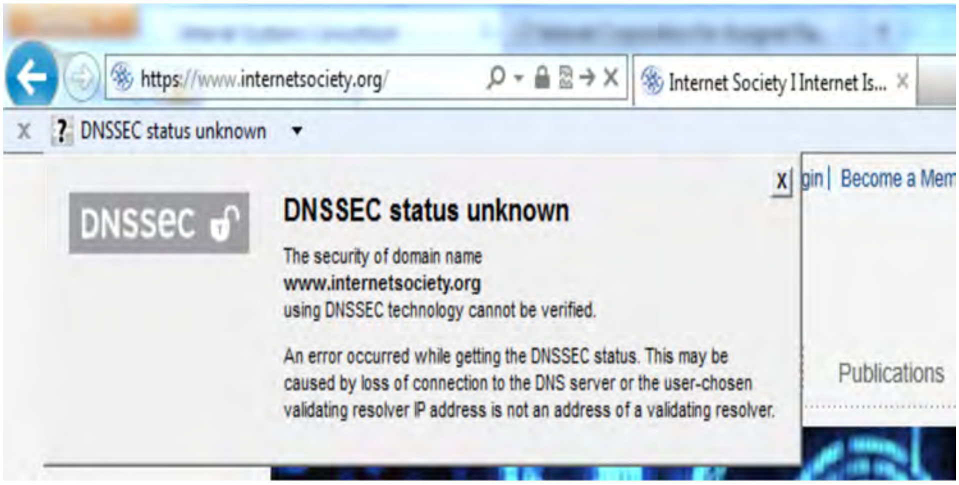
Figure 5.9: Mozilla FireFox DNSSEC Validator Settings

In order to determine the correct behavior of the web browser validators, all the three options were selected and tested separately. These 3 different options where used to build three test cases namely unsigned, signed remote and signed local resolvers. The test results for these test cases are presented below.

5.3.1 Unsigned Resolvers

This test was done using ISP provided Internet connection with unsigned name servers. The option use system settings was selected in DNSSEC validator preferences. This negative test was to establish if the web browser validators can correctly detect DNS responses without security data. Websites from the pool of signed, misconfigured and unsigned domains was used on the three web browsers using Windows 7 and Ubuntu 12.04 client platforms. The validators for all web browsers correctly established that the resolvers they were using were not security aware and DNSSEC status of domains visited could not be verified. In addition, they were also able to distinguish between signed and unsigned domains.

The web based validator is not able to determine the DNSSEC status of a domain if the it uses local unsigned resolvers as shown in Figure 5.10. The DNSSEC icon is displayed


https://www.internetsociety.org/
Internet Society I Internet Is...
DNSSEC status unknown
DNSSEC
gin | Become a Mem
Publications

DNSSEC status unknown

The security of domain name
www.internetsociety.org
using DNSSEC technology cannot be verified.

An error occurred while getting the DNSSEC status. This may be caused by loss of connection to the DNS server or the user-chosen validating resolver IP address is not an address of a validating resolver.

Figure 5.10: Validator using unsigned resolvers

in grey color indicating that the validator has failed to verify the DNSSEC status. The validator also provides an error message to the user that elaborates possible causes of the error.

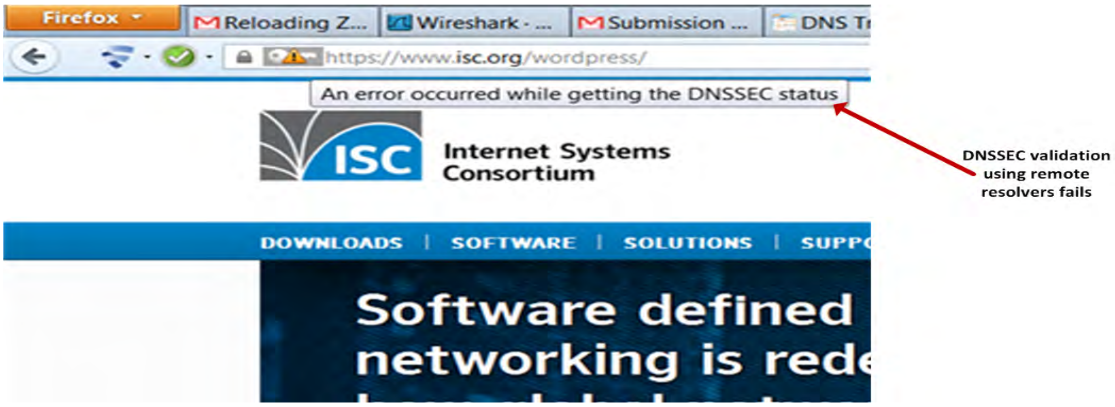
5.3.2 Signed Public Resolvers

In this test case, web browser validators were configured to use public DNSSEC validating name servers at CZ.NIC and OAR. The validators could not consistently determine the security status of domains visited through the web browsers. During the tests, on 12 out of 15 occasions, the validators correctly determined the security status of the domains being accessed. On 3 out of 15 occasions, the validators failed to validate the DNSSEC status of signed, unsigned and misconfigured domains. They displayed an error message shown in Figure 5.11. This means client computers using the remote DNSSEC validating resolvers cannot depend on these resolvers for DNS security all the time.

However, when available, the remote resolvers correctly determined the security status of some domains that were accessed via the web browsers.

5.3.3 Local Validating Resolvers

Unbound and DNSSEC Trigger were installed on Ubuntu 12.04 and Windows 7 client machines respectively. DNSSEC validation was enabled in the local resolvers. Web browser validators were configured to use the local resolvers by setting validator preference to 127.0.0.1, which refers to local resolvers on the computer. The web browsers were used


Firefox
Reloading Z...
Wireshark - ...
Submission ...
DNS Tr
https://www.isc.org/wordpress/
ISC
Internet Systems Consortium
DOWNLOADS | SOFTWARE | SOLUTIONS | SUPP
Software defined networking is rede

An error occurred while getting the DNSSEC status

DNSSEC validation using remote resolvers fails

Figure 5.11: Remote Resolver Error Message

to access signed, unsigned and misconfigured zones and DNSSEC validation results were observed and recorded.

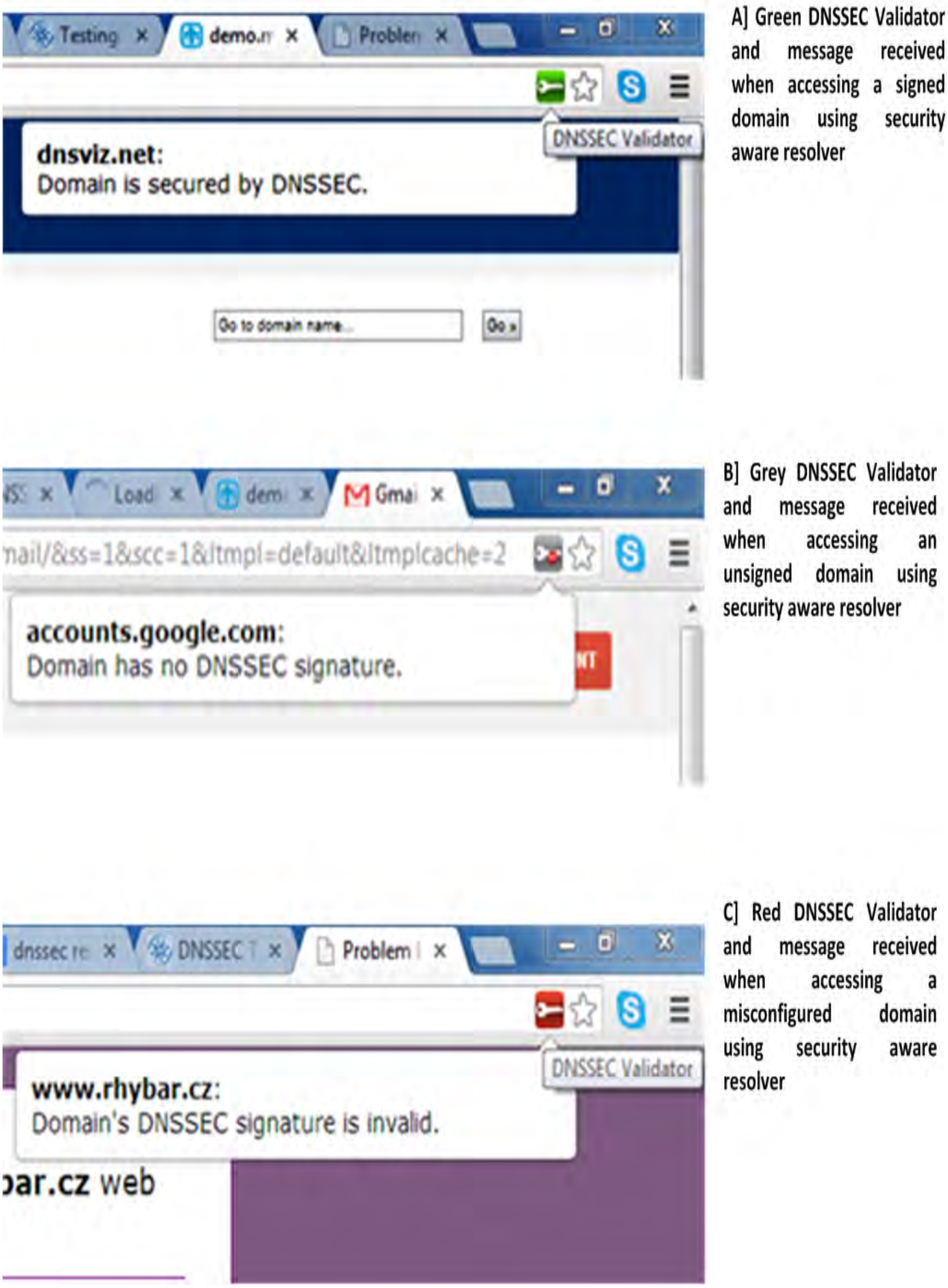
Web browser validators on the two client computers correctly validated signed domains and communicated the message through the DNSSEC indicator and corresponding messages. They also correctly detected domains with misconfigured signatures as shown in Figure 5.12. Unsigned domains were accurately detected as well. It was however noted that when using Unbound, there is need to configure it to allow web pages from unsigned domains. By default, it will display only web pages from signed domains.

Table 5.5 summarizes the web browsers and the resolver did not display the reason why the page is not available. This means that, if not properly configured, Unbound can cause unavailability of web resources from unsigned zones. Moreover, the reasons stated for the unavailability are not accurate.

Table 5.5 analyzes the behavior of DNSSEC web-based validators when different settings are selected. The behavior is tested against the pool of test domains tabled in Section 5.2. The results in the table show that when using a signed local resolver, all domains accessed are correctly validated. Signed remote resolvers correctly validated all queries but they have a 20% rate of validation failure, based on test results in Section 5.3.2.

5.4 Message Size

In addition to unsigned DNS query responses, security aware name servers also send cryptographic resource record signatures when responding to DNS queries. This increases the size of DNS messages handled by security aware resolvers. In determining the extent to which the message sizes increase, measurement and comparison of DNSSEC and non-



dnsviz.net:
Domain is secured by DNSSEC.

A] Green DNSSEC Validator and message received when accessing a signed domain using security aware resolver

accounts.google.com:
Domain has no DNSSEC signature.

B] Grey DNSSEC Validator and message received when accessing an unsigned domain using security aware resolver

www.rhybar.cz:
Domain's DNSSEC signature is invalid.

C] Red DNSSEC Validator and message received when accessing a misconfigured domain using security aware resolver

Figure 5.12: Google Chrome using Local Resolver

DNSSEC query responses was done.

DNSSEC was disabled in Unbound resolver in a Ubuntu virtual machine running on a

Table 5.5: Browser Validators with Local Resolvers

Domain/Resolver	**Unsigned domain**	**Mis-configured domain**	**Signed domain**
Non-validating Resolver	DNSSEC status is not verified. Validator icon is grey in color.	Domain is seen as signed but status cannot be verified. Validator icon is orange	Domain is identified as signed but status cannot be verified. Validator icon is orange.
Validating Local Resolver	DNSSEC status is not verified. Validator icon is grey in color.	DNSSEC status correctly identified as invalid. Validator icon is red	DNSSEC status identified as valid. Validator icon is green
Validating Remote Resolver	Validator identifies the domain as unsigned. Validator icon is grey.	DNSSEC status is identified as invalid. Validator icon is red.	DNSSEC status is identified as valid. Validator color is green

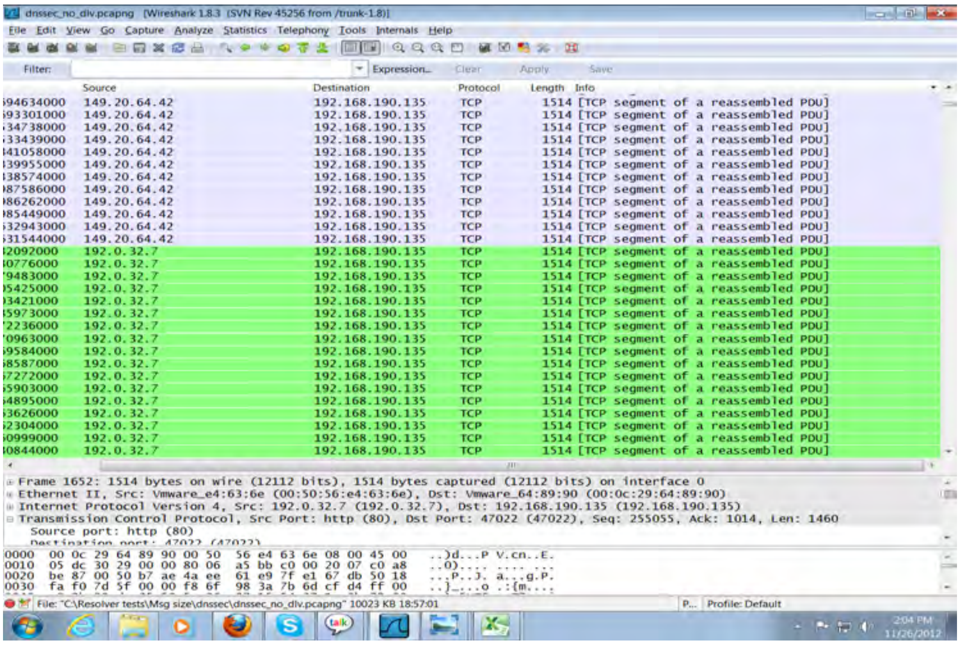
Windows 7 host. To obtain measurements of DNS messages, Wireshark was configured to capture all the network traffic on the Ubuntu virtual machine. DNS queries were made using web browsers to both signed and unsigned domains. The virtual machine network data file was saved in CSV format for analysis later.

After measuring the normal DNS message size, the resolver settings were configured to allow DNSSEC validation. The Ubuntu DNS cache was flushed to ensure that there are no cached DNS responses. Wireshark was set to capture all the network traffic data in a new file. Web browsers were used to access websites from signed domains only. The message analysis report was saved to a separate CSV file.

5.4.1 Message Fragmentation

Analysis of the data obtained from the tests show that there was fragmentation of DNSSEC messages that were over 1514 bytes. DNS query responses for security oblivious resolver did not exceed 1514 bytes and were not fragmented. Figure 5.13 presents a snapshot of fragmented packets from *www.isc.org* (146.20.64.42) and *www.icann.org* (192.0.32.7). The two are part of the signed domains that were used for this test.

As mentioned in Section 2.3.3, there is need to configure networking equipment to allow transportation of large DNSSEC packets. Internet users are mostly likely to be affected by such size limitation as they normally do not have administrative access to network equipment provided by service providers. Intranet users can have the networking equipment



```
dnssec_no_div.pcapng [Wireshark 1.8.3 (SVN Rev 45256 from /trunk-1.8)]
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Tools  Internals  Help
Filter:                                      Expression...  Clear  Apply  Save
         Source                   Destination       Protocol  Length Info
94634000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
93301000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
34738000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
33439000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
41058000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
39955000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
38574000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
87586000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
86262000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
85449000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
32943000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
31544000 149.20.64.42             192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
2092000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
0776000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
9483000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
5425000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
3421000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
5973000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
2236000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
0963000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
9584000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
8587000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
7272000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
5903000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
4895000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
3626000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
2304000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
0999000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]
0844000  192.0.32.7               192.168.190.135   TCP       1514 [TCP segment of a reassembled PDU]

⊞ Frame 1652: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
⊞ Ethernet II, Src: Vmware_e4:63:6e (00:50:56:e4:63:6e), Dst: Vmware_64:89:90 (00:0c:29:64:89:90)
⊞ Internet Protocol Version 4, Src: 192.0.32.7 (192.0.32.7), Dst: 192.168.190.135 (192.168.190.135)
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: 47022 (47022), Seq: 255055, Ack: 1014, Len: 1460
    Source port: http (80)
    Destination port: 47022 (47022)

0000  00 0c 29 64 89 90 00 50  56 e4 63 6e 08 00 45 00   ..)d...P V.cn..E.
0010  05 dc 30 29 00 00 80 06  a5 bb c0 00 20 07 c0 a8   ..0).... .... ...
0020  be 87 00 50 b7 ae 4a ee  61 e9 7f e1 67 db 50 18   ...P..J. a...g.P.
0030  fa f0 7d 5f 00 00 f8 6f  98 3a 7b 6d cf d4 ff 00   ..}_...o .:{m....

File: "C:\Resolver tests\Msg size\dnssec\dnssec_no_div.pcapng" 10023 KB 18:57:01    P...    Profile: Default
2:04 PM
11/26/2012
```

Figure 5.13: DNSSEC Packet fragmentation

reconfigured to allow such packets since organizations normally have their own networking equipment

5.4.2 Standard Queries

Figure 5.14 shows the distribution of data packets for standard DNS queries for security aware and security oblivious resolvers. The packets bundled into groups of 20 bytes each for easier analysis. The graph shows that most of the standard DNS queries from a security oblivious resolver fell into the 61-80 bytes group. It also shows that most of DNS queries from a security aware resolver fell into the 81-100 bytes group. Furthermore, the largest query packet recorded for a non-DNSSEC resolver was at most 100 bytes whilst the largest packet from a DNSSEC resolver falls into the 121-140 byte group. This shows that standard queries emanating from a security aware resolver are significantly larger than those from a security oblivious resolver. The increase in the size of DNS queries from security aware resolvers is caused by the additional bits that indicate that a resolver is security aware as explained in Section 3.2.2.

5.4.3 Query Responses

The DNS packets recorded in the experiment show that there was a huge difference in the size of data packets received as DNS query responses for DNSSEC and non-DNSSEC

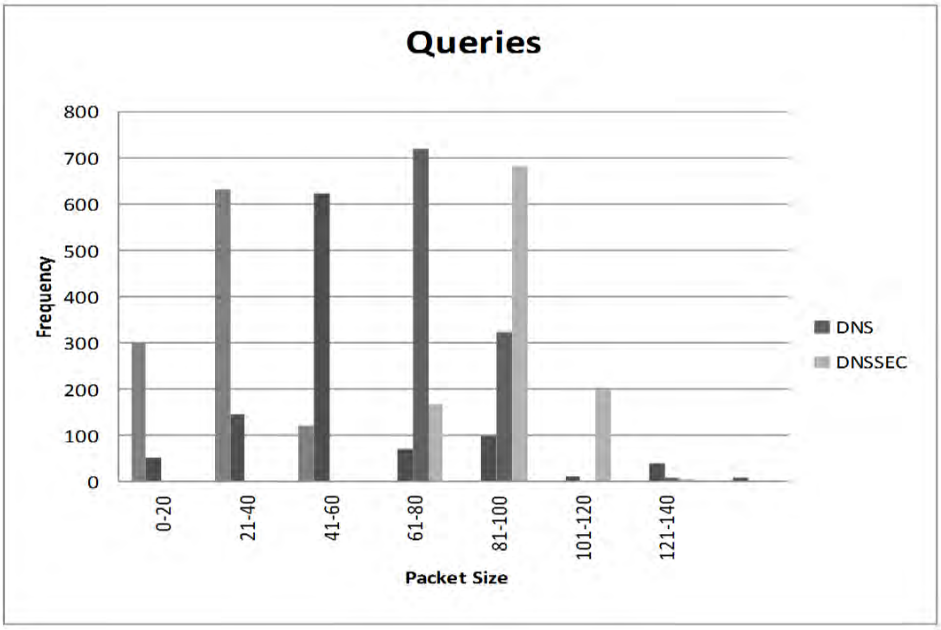


Figure 5.14: Standard DNS Queries

resolvers. The maximum data packet received by a non-DNSSEC resolver was 516 bytes in size whilst the maximum received by DNSSEC resolver was 1514 bytes in size. Figure 5.15 show the results and the differences in data packets for DNS and DNSSEC queries.

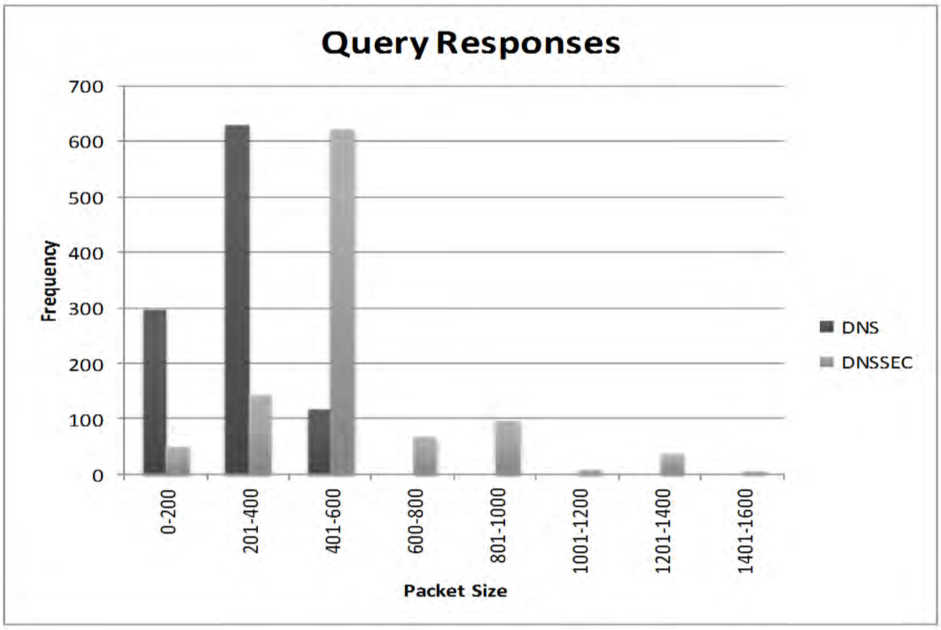


Figure 5.15: Query Responses

Further analysis of query responses show that security oblivious resolvers only receive final address or NXDOMAIN answers for their queries. On the other hand, signed resolvers receive normal responses with their corresponding RRSIG. The RRSIGs account for the difference in message sizes between signed and unsigned resolvers.

5.5 Summary

The design of the experimental environment is also described in detail in this chapter. Results of tests done on DLV, DNSSEC validating resolvers, Web based validators and message size are presented and interpreted. The results will be used to verify if research objectives were met in the next chapter.

Chapter 6

Conclusion

This chapter looks at how this thesis has achieved the research objectives through answering the research questions set in the beginning of the thesis. It also looks at further studies that can be undertaken to promote continued deployment of DNSSEC.

6.1 Research Questions

The thesis provided answers to the research questions outlined in Section 1.4. The summarized answers to the research questions outlined in Section 1.4 are presented here.

Research Question 1: "What does it take to deploy DNSSEC in islands of security?"

Deploying DNSSEC in islands of security can be done on a DNS client level or server level. Deploying DNSSEC at client level involves installing DNSSEC validating resolvers such as DNSSEC Trigger and Unbound as detailed in Section 4.3. These are able to validate the authenticity and integrity of DNS response coming from signed domains. Web based validators add visibility of DNSSEC validation results by distinguishing between validated and unvalidated results with different colors on the web browser as illustrated in Section 5.3. Although DNSSEC validating applications offer the best security for end users when combined with validating resolvers, they are not widely available at the moment. Both Internet and intranet users can make use of this solution to mitigate against DNS security threats.

On the server side, organizations that operate in islands of security can deploy DNSSEC by signing their zone and publishing their trust anchors through DLV repositories such as ISC as presented in Section 5.1. This will enable security aware resolvers to be able to determine authenticity and integrity of their responses. This will also provide security assurance to users accessing resources in their domain.

Research Question 2: “What are the benefits of DNSSEC for an average user?”

Once a user connects to the Internet, domain name resolution is used to access the required resource. DNSSEC introduces a way of verifying the authenticity and integrity of DNS messages to mitigate against sending spoofed responses to users as explained in Section 3.1. DNSSEC gives the end user assurance that they are connecting to the correct domain when using the Internet. It also verifies that responses from authoritative name servers have not been modified. This helps protect users against phishing and other attacks outlined in Section 2.5 when they are using the Internet.

At this stage where DNSSEC is not widely deployed, users with suitably configured DNSSEC validating resolvers will be able to distinguish between validated and unvalidated responses. Web browsers with DNSSEC validators show a warning message when a web service being accessed is from an unsigned domain as shown in Section 5.3. They also show a message when the resolver fails to validate the resource record signatures due to misconfiguration or forging of responses. This is also helps bring security awareness issues to the user as the message explains the risk associated with accessing a web service from an unsigned domain.

However, end users stand to obtain greater benefits from DNSSEC when more organizations deploy DNSSEC. As it stands, most domains are not signed and end users who use the available DNSSEC tools are still exposed to the security risks associated with unsigned domains. It is therefore important that organization offering web based services deploy DNSSEC for end users to obtain greater benefits from using web based validators and security aware resolvers such as DNSSEC Trigger.

Research Question 3: “What are the DNSSEC tools and applications available?”

DNSSEC applications can be broadly grouped into authoritative name servers and recursive resolvers. The name server software supporting applications have increased. This research used Bind 9.8 for deploying DNSSEC. However there are other DNS server applications that are DNSSEC enabled. The most commonly used are listed in table 6.1. Organizations planning to deploy DNSSEC can also choose from

Table 6.1: DNSSEC Software

Software	Description	Developers
Bind	Authoritative and recursive DNS server with support for DNSSEC.	ISC
Windows Server 2008 R2	Authoritative and recursive DNS server with limited DNSSEC support.	Microsoft
Unbound	Recursive DNS server with DNSSEC support.	NLNetlabs
DNSSEC Trigger	Recursive.	NLnetlabs
Knot DNS	Authoritative DNSSEC support DNS server.	CZ.NIC

The signing of the zone used for the purposes of this research was made easier with the open source tools that are available for open source DNS applications. These tools include zonesigner, rollerd and donuts. Bind 9.8 used in this research comes with a package called dnssec-tools that is automatically installed with the application. It contains these tools which simplify the tasks involved in zone signing and key rollover. Table 6.2 provides details of some of the tools and their functionality.

Table 6.2: DNSSEC Tools

Tool	Description	Developers
Zonesigner	Tool that is used for signing a zone and generating keys.	SPARTA, Inc
Rollerd	Open source key rollover tool.	SPARTA, Inc
Donuts	Open Source tool for checking errors in signed zones.	SPARTA, Inc
named-checkzone	A tool for testing Bind zones for errors.	ISC
DNSSEC Debugger	Online DNSSEC testing and debugging tool.	Verisign

Research Question 4: "What is the impact of DNSSEC on DNS traffic and message size?"

DNSSEC introduces new resource records to DNS as explained in Section 3.1. The DNSSEC message size tests presented in Section 5.4 show that the difference between largest DNS query emanating from security aware is 40 bytes bigger than that coming from security oblivious resolvers. However, there are bigger differences between query responses with security data and those without security data as shown on Figure 5.17 . The increase in response size is caused by RRSIGs that are sent together with the normal resource records when a signed name server is responding to security aware resolvers.

The increased DNSSEC message size results in message truncation illustrated in Figure 5.4.1. Furthermore, large DNS packets are transported using TCP which is much slower than UDP. This results in an increase in query turnaround time.

Research Question 5: "What are the costs involved in implementing DNSSEC?"

All the software used to setup the DNSSEC environments in this thesis is open source and provide free of charge. Bind, Unbound and DNSSEC Trigger were downloaded from the internet free of charge. Web based validators were also downloaded from the internet free of charge. This means that setting up DNSSEC by both Internet and Intranet clients can be achieved at no cost at all in as far as software and zone management tools are concerned. Other DNSSEC software listed in table 6.1 like Windows Server 2008 R2 and Knot DNS are not open source and have to be paid for.

Whilst most of the DNSSEC software and tools are available free of charge, there are other indirect costs associated with deploying DNSSEC. For example zone key management introduces a new administrative cost because of the need for regular key rollover. Other indirect costs such as securing storage of the private keys can also introduce new costs. For upstream service providers such as domain registrars, signing all the zone records for all domains under their control increases storage requirements. Therefore, the costs of DNSSEC for end users and zone operators are less as compared to the costs that will be incurred by upstream service providers.

6.2 Research Objectives

The aims of this research were outlined in Section 1.2. In order to assess whether the research objectives were met, each objectives will be revisited and analyzed in the context of the results detailed in Chapter 5.

6.2.1 Methods of deploying DNSSEC

One of the objectives of this research was to investigate the methods that can be used to deploy DNSSEC in islands of security. The tests done in Chapter 5 evaluated one server side and two client side methods of deploying DNSSEC by organizations and users whose parent domain is not signed. The methods tested are DLV, DNSSEC validating resolvers and web based validators. All the three methods proved useful for deploying DNSSEC.

As mentioned in Section 5.3.3, there is need to combine local resolvers with web based validators for improved validation of signed responses at the client side. This is because web based validators are more effective when using a local resolver. In addition, web based validators provide information on the DNSSEC status of a domain they have visited. Resolvers on their own do not provide that information. This model of combining local resolvers and web based validators will be more effective to Internet Clients (A and B in Section 3.4).

The zone demo.moria.org could supply security data in response to queries after it was signed and registered with the ISC DLV registry as shown in Section 5.1.2. Therefore, DLV can be used to deploy DNSSEC by organizations willing to provide their clients with security. This can provide a competitive edge in business. The process of managing zone keys need to be carefully planned to avoid domain unavailability due to expired keys (Chandramouli and Rose, 2010). The process of uploading zone keys with ISC DLV adds an extra chunk of administrative work as explained in Section 5.1. Care should also be taken to increment the zone serial number every time keys are changed so that zone changes are propagated to other caching resolvers.

6.2.2 Recommendations

Section 3.4 identified two groups of users namely Internet and Intranet users. The recommendations on how to deploy DNSSEC will be detailed for each group of users based on the results of the tests done in Chapter 5.

1. **Internet Clients** - Local resolvers combined with web based validators will provide visible security against most of the DNS security threats mentioned in Section 2.4. The local resolver will validate DNS query response received for security while the web based validator will provide a way of visualizing the DNSSEC status of the domain visited. Local resolvers solve the security problem of the channel of communication between a remote resolver and the client computer. Remote DNSSEC

validating resolvers can be used with web based validators but the test results in Section 5.3 show that remote resolvers are not available all the time. This can cause validation failure. Although the client side DNSSEC validating applications are open source and available free of charge, downloading them and and installing can bring in an additional attack vector for an average internet user. Furthermore, there may be need to educate internet users on how the validators and local resolvers work.

2. **Intranet Clients** - In addition to the need for users in a corporate network to access validated DNS responses through DNSSEC aware resolvers, corporates also have a responsibility to provide security to the public accessing their domains (Lowinder *et al.*, 2010). This can be achieved by deploying DNSSEC through DLV until the TLD is signed. The results of this research presented in Section 5.1.2 show that demo.moria.org issues DNSSEC responses with security data. Based on these results DLV can be depended on for deploying DNSSEC by organizations in islands of security. Bind helps to make the process of managing a zone much easier as it is bundled with open source zone management tools such as zonesigner, rollerd and donuts mentioned in Section 4.3. Key rollover periods should be balanced to reduce administrative work associated with key rollover whilst maintaining security. Experience obtained in managing the demo.moria.org suggest a longer key rollover period to reduce the need to resign the zone after short periods of time. A key rollover of 60 days will attempt to balance the need to maintain security and reducing zone administration tasks.

6.2.3 Slow DNSSEC uptake

The rate of DNSSEC is slow especially in Africa as outlined in Section 3.4.1. From the experiences in signing the zone demo.moria.org, two possible reasons were identified for the slow uptake. Using DLV for DNSSEC deployment increases the zone management work required. In addition to periodical key rollover on the actual name server, there is need to manage the keys on the DLV domain as presented in Section 5.1. One also needs to keep private keys in a secure way away from the physical server. In addition, if any mistakes are made the domain will be unavailable until the TTL expires.

Another possible reason why DNSSEC is not widely deployed is the increased message sizes. The networking equipment may be running on legacy software and need to be upgraded to allow large DNS packets. Some DNS servers are configured not to allow certain aspects of DNSSEC as shown in Section 5.1.1. This is a challenge particularly for organizations that do not host their own domains. These issues, combined with the extra administrative work can be seen to be increasing costs for organizations and could be stifling the widespread deployment of the security extensions to the DNS.

Other reasons for slow deployment of DNSSEC are outlined in Section 3.7 of this thesis.

6.2.4 Effect of DNSSEC to DNS message size

The test results presented in Sections 5.4.1 and 5.4.2 show that DNSSEC increases the size of DNS messages. The large DNS packets are truncated by legacy networking equipment in use in many networks (Bellis and Phifer, 2008), (Eklov *et al.*, 2010). For example, the largest DNSSEC packet allowed in Section 5.4 was 1514 bytes through TCP. TCP is much slower than UDP, hence the message transportation will be slow. Therefore DNSSEC has a negative effect on DNS message sizes.

6.3 Future Work

One of the major drawbacks of DNSSEC is that it results in an increase in DNS message sizes. The increase in DNS message sometimes causes message fragmentation as observed in Section 5.4.1. Cases of DNS message failure due to increased sizes have also been recorded. Some networking equipment need to be adjusted to allow DNS messages of over 512 bytes in size.

Whilst DNSSEC is being implemented to mitigate against DNS security threats outlined in Section 2.4 of this work, IPV6 is being implemented to address the impending shortage of global IP address brought about by increase in computing and related machinery such as smart phones and tablets requiring IP addresses.

IPv6 will however increase the size of IP addresses from 32 bit currently in use to 128 bits. If all computers and related devices on the internet move over to IPv6, this will significantly increase the size of DNS messages. It is therefore important to determine the effect, if any of IPv6 on DNS message sizes so that corrective action can be taken in advance if IPv6 is going to have any effect on DNS message sizes.

References

Aitchison, R. *Pro DNS and Bind*. Edition 1,ISBN 1-59059-494-0. Apress, August 2005.

Alexiou, N., Basagiannis, S., Deshpande, T., Smolka, S. A., and Katsaros, P. *Formal Analysis of the Kaminsky DNS Cache-Poisoning Attack Using Probabilistic Model Checking. IEEE 12th International Symposium on High-Assurance Systems Engineering (HASE), 2010*, pages 94 – 103, 2010. Last Accessed on 12 February 2013.
URL http://users.auth.gr/basags/pubs/hase9.pdf

Ali, Y. and Smith, S. W. *Flexible and Scalable Public Key Security for SSH.* In *EuroPKI*, pages 43–56. 2004. Last Accessed on 12 February 2013.
URL http://www.cs.dartmouth.edu/~sws/pubs/as04.pdf

Andersson, K. and Montag, D. *Development of DNS security, attacks and countermeasures*. Internet, 2008. Last Accessed on 12 February 2013.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.7896&rep=rep1&type=pdf

Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S. *IETF RFC 4033. DNS Security Introduction and Requirements*, May 2005a. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc4033.txt

Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S. *IETF RFC 4034. Resource Records for the DNS Security Extensions*, 2005b. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc4034.txt

Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S. *IETF RFC 4035. Protocol Modifications for the DNS Security Extensions*, March 2005c. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc4035.txt

Ariyapperuma, S. and Mitchell, C. J. *Security vulnerabilities in DNS and DNSSEC. IEEE: The Second International Conference on Availability, Reliability and Security 2007*, 2nd:Page(s): 335 – 342, 2007. Last Accessed on 12 February 2013.
URL http://isg.rhul.ac.uk/~cjm/svidad.pdf

Atkins, D. and Austein, R. *IETF RFC 3833. Threat Analysis of the Domain Name System (DNS)*, August 2004. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc3833.txt

Babu, P. R., Bhaskari, D., and CH.Satyanarayana. *A Comprehensive Analysis of Spoofing. (IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 1,No.6, December 2010. Last Accessed on 12 February 2013.
URL http://thesai.org/Downloads/Volume1No6/Paper_23_A_Comprehensive_Analysis_of_Spoofing.pdf

Barnes, R. L. *DANE: Taking TLS Authentication to the Next Level Using DNSSEC. IETF Journal October 2011*, 7 Issue 2, 2011. Last Accessed on 12 February 2013.
URL http://www.internetsociety.org/articles/dane-taking-tls-authentication-next-l

Bau, J. and Mitchell, J. C. *A Security Evaluation of DNSSEC with NSEC3. NDSS Symposium*, 2010. Last Accessed on 14 October 2012.
URL http://crypto.stanford.edu/~jcm/papers/dnssec_ndss10.pdf

Bellis, R. and Phifer, L. *Test Report:DNSSEC Impact on Broadband Routers and Firewalls. Nominec and Core Competence*, 2008. Last Accessed on 13 February 2013.
URL http://download.nominet.org.uk/dnssec-cpe/DNSSEC-CPE-Report.pdf

Bernstein, D. J. *High-speed high-security cryptography: encrypting and authenticating the whole Internet. 27th Chaos Communication Congress*, 2010. Last accessed on 24 October 2012.
URL http://events.ccc.de/congress/2010/Fahrplan/attachments/1773_slides.pdf

Chandramouli, R. and Rose, S. *Secure Domain Name System (DNS) Deployment Guide*. Internet, April 2010. Last Accessed on 12 February 2013.
URL http://csrc.nist.gov/publications/nistpubs/800-81r1/sp-800-81r1.pdf

Conrad, D. *IETF RFC 3225. Indicating Resolver Support of DNSSEC*, August 2001. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc3225.txt

Dagon, D., Antonakakis, M., Day, K., Luo, X., Lee, C. P., and Lee, W. *Recursive dns architectures and vulnerability implications.* In *Network and Distributed System Security Symposium (NDSS09)*. 2009. Last Accessed on 12 February 2013.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.217.2671&rep=rep1&type=pdf

Dagon, D., Antonakakis, M., Vixie, P., Jinmei, T., and Lee, W. *Increased DNS forgery resistance through 0x20-bit encoding: security via leet queries.* In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 211–222. ACM, New York, NY, USA, 2008. ISBN 978-1-59593-810-7. doi:10.1145/1455770.1455798. Last Accessed on 24 October 2012.
URL http://doi.acm.org/10.1145/1455770.1455798

Daigle, L. *RFC Editor in Transition: Past, Present, and Future. The Internet Protocol Journal - Volume 13, Number 1*, 13:26–32, 2010. Last Accessed on 12/02/2012.
URL http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_13-1/131_rfc.html

Davies, K. *2008 DNS Cache Poisoning Vulnerability. ICANN, Cairo 2008*, 2008. Last Accessed on 12 October 2012.
URL http://www.iana.org/about/presentations/davies-cairo-vulnerability-081103.pdf

Deutsch, L. P. *IETF RFC 606. Host Names On-line*, 1973. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc606.txt

Dierks, T. and Rescorla, E. *IETF RFC 5246. The Transport Layer Security (TLS) Protocol Version 1.2*, 2008. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/html/rfc5246

DNS-OARC. *Oarc's open dnssec validating resolvers.* Internet, June 2011. Last Accessed on 30/03/2013.
URL http://www.dns-oarc.net/oarc/services/odvr

DNS-OARC. *DLV Test Zones.* Internet, August 2012a. Last Accessed on 12 February 2013.
URL https://www.dns-oarc.net/oarc/services/dlvtest

DNS-OARC. *Mitigating DNS Denial of Service Attacks*. Internet, March 2012b. Last Accessed on 12 February 2013.
URL https://www.dns-oarc.net/wiki/mitigating-dns-denial-of-service-attacks

DNS-OARC. *Number of TLD's with DS records*. Internet, June 2012c. Last Accessed on 12 February 2013.
URL https://www.dns-oarc.net/oarc/data/zfr/root/ds

DNSSEC Deployment Initiative. Internet, 2009. Last Accessed on 02 October 2012.
URL https://www.dnssec-deployment.org/wp-content/uploads/2010/06/TLD-deployment-Table1.pdf

DNSSEC Deployment Initiative. *DNSSEC at FOSE 2012*. Internet, 2012. Last Accessed on 12 February 2013.
URL https://www.dnssec-deployment.org/index.php/presentations-events-and-newsletters/dnssec-at-fose-2012/

Dnssec Deployment Initiative. *Tools and Resources*. Internet, February 2012. Last Accessed on 12 February 2013.
URL https://www.dnssec-deployment.org/wiki/index.php/Tools_and_Resources

DNSViz. *A DNS Visualization tool*. Internet, February 2013a. Last Accessed on 12 February 2013.
URL htttp://dnsviz.net/d/demo/moria.org/dnssec

DNSViz. *A DNS visualization tool*. Internet, February 2013b. Last Accessed on 12 February 2013.
URL dnsviz.net/d/demo.moria.org/servers

DNSViz. *DNS Server Responses*. Internet, February 2013c. Last Accessed on 12 February 2013.
URL http://dnsviz.net/d/demo.moria.org/responses/

Eastlake, D., Schiller, J., and Crocker, S. *IETF RFC 4086. Randomness Requirements for Security*, 2005. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc4086.txt

Edge, J. *DNSCurve: an alternative to DNSSEC*. July 2009. Last Accessed on 12 February 2013.
URL http://lwn.net/Articles/340528/

EDUCAUSE. *7 Things You Should Know About DNSSEC*. 2010. Last Accessed on 12 February 2013.
URL http://net.educause.edu/ir/library/pdf/EST1001.pdf

Eklov, T., Gefle, I., and Lagerholm, S. *Implementing DNSSEC*. *The Internet Protocol Journal, Volume 13, No.2*, 13, 2010. Last Accessed on 12 February 2013.
URL http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_13-2/132_dnssec.html

Eklov, T. and Lagerholm, S. *Implementing DNSSEC*. *The Internet Protocol Journal - Volume 7, Number 2*, 7:16–26, 2010. Last Accessed on 12 February 2013.
URL http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_13-2/ipj_13-2.pdf

Eland, H. *Securing a Domain: SSL vs. DNSSEC*. *Afilias Blog*, 2009. Last Accessed on 24 October 2012.
URL http://afilias.info/webfm_send/32

ENISA. *The costs of dnssec deployment*. Internet, November 2009. Last Accessed on 12 February 2013.
URL http://www.enisa.europa.eu/activities/res/technologies/tech/dnsseccosts

EURid. *Overview of DNSSEC deployment worldwide*. Internet, October 2010. Last Accessed on 24 October 2012.
URL http://www.eurid.eu/files/Insights_DNSSEC1.pdf

Filip, O. *DNSSEC.CZ*. *ICANN 44,Prague 2012.*, 2012. Last Accessed on 24 October 2012.
URL http://prague44.icann.org/meetings/prague2012/presentation-dnssec-cz-27jun12-en.pdf

Gieben, M. *DNSSEC: The Protocol, Deployment, and a Bit of Development*. *The Internet Protocol Journal - Volume 7, Number 2*, 2004. Last Accessed on 24 October 2012.
URL http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-2/dnssec.html

Granstrom, M. *What is wrong with DNS?* *TKK T-110.5190 Seminar on Internetworking*, 2009. Last Accessed on 24 October 2012.

URL http://www.cse.tkk.fi/en/publications/B/5/papers/Granstrom_final.pdf

Green, I. *DNS Spoofing by The Man In The Middle. Sans InfoSec Reading Room*, 2005. Last Accessed on 24 October 2012.

Gudmundsson, O. *IETF RFC 3658. Delegation Signer (DS) Resource Record (RR)*, 2003. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/rfc/rfc3658.txt

Hardaker, W. and Krishnaswamy, S. *Enabling DNSSEC in Open Source Applications. Securing and Trusting Internet Names, SATIN 2012*, 2012. Last Accessed on 24 October 2012.
URL http://conferences.npl.co.uk/satin/papers/satin2011-Hardaker.pdf

Harrenstien, K., Stahl, M., and Feinler, E. *IETF RFC 952. DOD INTERNET HOST TABLE SPECIFICATION*, October 1985. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc952.txt

Huston, G. *Counting DNSSEC*. Internet, September 2012. Last Accessed on 12 February 2013.
URL https://labs.ripe.net/Members/gih/counting-dnssec

IANA. *Domain Name System Security (DNSSEC) Algorithm Numbers*. Internet, July 2012a. Last Accessed on 24 October 2012.
URL http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xml

IANA. *Root Servers*. Internet, October 2012b. Last Accessed on 24 October 2012.
URL http://www.iana.org/domains/root/servers

ICANN. *SSAC Advisory SAC008 DNS Distributed Denial of Service (DDoS) Attacks. ICANN Security and Stability Advisory Committe*, 2006. Last Accessed on 12 February 2013.
URL http://www.icann.org/en/committees/security/dns-ddos-advisory-31mar06.pdf

ICANN. *DNSSEC securing the Internet: Benefits to Companies and Consumers*. 2012a. Last Accessed on 24 October 2012.
URL http://www.icann.org/en/news/in-focus/dnssec/dnssec-card-03dec12-en.pdf

ICANN. *Internet Corporation for Assigned Names and Numbers*. Internet, October 2012b. Last Accessed on 24 October 2012.
URL http://www.icann.org/en/about/welcome

ICANN. *Map of DNSSEC TLD's*. Internet, 2012c. Last Accessed on 24 October 2012.
URL http://www.icann.org/en/news/in-focus/dnssec/deployment-tlds

ICANN. *Registrars that support end user DNSSEC management, including entry of DS records*. Internet, April 2012d. Last Accessed on 24 October 2012.
URL http://www.icann.org/en/news/in-focus/dnssec/deployment

ICANN. *TLD DNSSEC Report*. Internet, August 2012e. Last Accessed on 24 October 2012.
URL http://stats.research.icann.org/dns/tld_report/

ICANN. *TLD DNSSEC Report (2012-07-26)*. Internet, July 2012f. Last Accessed on 24 October 2012.
URL http://stats.research.icann.org/dns/tld_report/

Internet Society. *How To Secure And Sign Your Domain With DNSSEC Using Domain Registrars*. Internet, 2012. Last Accessed on 24 October 2012.
URL http://www.internetsociety.org/deploy360/resources/dnssec-registrars/

ISC. *DLV Policy Document*. 2009a. Last Accessed on 12 February 2013.
URL http://www.isc.org/files/dlv-policy.pdf

ISC. *DNSSEC Look-aside Validation*. May 2009b. Last Accessed on 12 February 2013.
URL https://www.dns-oarc.net/files/workshop-200905/mitchell.pdf

ISC. *DNS and DNSSEC Terminology*. 2010. Last Accessed on 12 February 2013.
URL https://dlv.isc.org/about/dnssec_records

ISC. *BIND 9 Administrator Reference Manual*. 2012. Last Accessed on 12 February 2013.
URL http://ftp.isc.org/isc/bind9/cur/9.7/doc/arm/Bv9ARM.pdf

Janssen, P. *.eu DNSSEC deployment*. *ICANN 44, Prague 2012*, 2012. Last Accessed on 12 February 2013.
URL http://prague44.icann.org/meetings/prague2012/presentation-eu-dnssec-deployment-27jun12-en.pdf

Kaminsky, D. *Black Ops 2008:It's The End Of The Cache As We Know It. Black Hat USA*, 2008. Last Accessed on 12 February 2013.
URL http://kurser.lobner.dk/dDist/DMK_BO2K8.pdf

Kaminsky, D. *DNSSEC Interlude 2: DJB@CCC. Dan Kaminsky's Blog*, 2011. Last Accessed on 12 February 2013.
URL http://dankaminsky.com/2011/01/05/djb-ccc/

Kaufman, C. and Eastlake, D. *IETF RFC 2065. Domain Name System Security Extensions*, January 1997. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc2065.txt

Kent, S. and Atkinson, R. *IETF RFC 2401. Security Architecture for the Internet Protocol*, November 1998. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc2401.txt

Kolkman, O. *DNSSEC HOWTO, a tutorial in disguise. NLnet Labs*, pages 50–54, 2009. Last Accessed on 24 October 2012.
URL http://www.nlnetlabs.nl/dnssec_howto/dnssec_howto.pdf

Kolkman, O. and Gieben, R. *IETF RFC 4641. DNSSEC Operational Practices*, 2006. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc4641.txt

Kolkman, O. M. *DNSSEC Advantage: Reasons for deploying DNSSEC.* Internet, September 2008. Last Accessed on 24 October 2012.
URL http://www.dnssec.net/why-deploy-dnssec

Kos, A., Zwittnig, B., Kozic, D., and Sterle, J. *DNSSEC Key Management. E LEKTROTEHNISKI VESTNIK*, 79(1-2):47–54, 2012. Last Accessed on 24 October 2012.
URL http://ev.fe.uni-lj.si/1-2-2012/Kozic.pdf

Kristoff, J. and Joffee, R. *Botnets and Packet Flooding DDoS Attacks on the Domain Name System. International Journal of Forenscic Computer Science-IJoFCS*, 2:9–18, May 2007. Last Accessed on 24 October 2012.
URL http://layer9.com/~jtk/papers/dnsddos.pdf

Kudlick, M. *IETF RFC 608. HOST NAMES ON-LINE*, January 1974. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/html/rfc608

Kurose, J. F. and Ross, K. W. *Computer Networking: A Top-Down Approach.* Addison-Wesley Publishing Company, USA, 5th edition, 2009. ISBN 0136079679, 9780136079675. Last Accessed on 24 October 2012.

Laurie, B., G.Sisson, Arends, R., and Blacka, D. *IETF RFC 5155. NSEC3*, 2008. Last Accessed on 12 February 2013.
URL www.ietf.org/rfc/rfc5155.txt

Lawrence, D. *IETF RFC 3425. Obsoleting IQUERY*, November 2002. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/rfc/rfc3425.txt

Lewis, E. *IETF RFC 3130. Notes from the State-Of-The-Technology: DNSSEC*, 2001. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/html/rfc3130

Lioy, A., Mazzocchi, D., Maino, F., and Marian, M. *DNS Security. Terena Networking Conference, May 22-25, 2000*, May 2000. Last Accessed on 24 October 2012.
URL https://www.terena.org/events/archive/tnc2000/proceedings/3A/3a3.pdf

Liu, C. and Albitz, P. *DNS and Bind.* ISBN 1-56592-512-2. O'Reilly Media, 1998.

Lottor, M. *IETF RFC 1033. Domain Administrators operations guide*, November 1987. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc1033.txt

Lowinder, A. E., Zacharopoulos, D., Ljunggren, F., Donnerhacke, L., OKolkman, Faltstrom, P., and Wallstrom, P. *Good Practices for deploying DNSSEC.* Internet, March 2010. Last Accessed on 24 October 2012.
URL http://www.enisa.europa.eu/activities/res/technologies/tech/gpgdnssec

Lowinder, A.-M. E. *Step by step DNSSEC deployment in .se. Securing and Trusting Internet Names, SATIN 2012*, March 2012. Last Accessed on 24 October 2012.
URL http://conferences.npl.co.uk/satin/presentations/satin2012slides-EklundLowinder.pdf

Massey, D., Osterweil, E., and Zhang, L. *Observations from the DNSSEC Deployment. 3rd IEEE Workshop on Secure Network Protocols, 2007. NPSec 2007*, 3:1–6,

October 2007. Last Accessed on 12 February 2013.
URL http://irl.cs.ucla.edu/papers/SecSpider_NPSec07.pdf

Microsoft. *Windows Server 2008 R2*. Internet, 2008. Last Accessed on 24 October 2012.
URL http://technet.microsoft.com/en-us/windowsserver/bb310558.aspx

Mockapetris, P. *IETF RFC 1034. Domain names - Concept and facilities*, November 1987a. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc1034.txt

Mockapetris, P. *IETF RFC 1035. Domain names - Implementation and specification*, November 1987b. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc1035.txt

Mohan, R. *Five Strategies for Flawless DNSSEC Key Management and Rollover*. Internet, July 2010. Last Accessed on 26 july 2012.
URL http://www.securityweek.com/five-strategies-flawless-dnssec-key-management-a

Mundy, R. *Beyond Infrastructure: Emerging DNSSEC Apps and DANE. Making DNSSEC the Trust Infrastructure: Where Domain Name Security is Headed, FOSE 2012*, 2012. Last Accessed on 24 October 2012.
URL https://www.dnssec-deployment.org/wp-content/uploads/2012/03/5-Mundy-Apps-RM1.ppt

Murali, U. *Three Reasons Why It Makes Sense to Deploy DNSSEC Now*. Internet, June 2010. Last Accessed on 24 October 2012.
URL http://www.circleid.com/posts/three_reasons_why_it_makes_sense_to_deploy_dnssec_now/

Neigus, N. and Feinler, J. *IETF RFC 597. Host Status*, 1973. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc597.txt

NIST. *Estimating IPv6 & DNSSEC Deployment SnapShots*. Internet, 2012. Last Accessed on 27 July 2012.
URL http://fedv6-deployment.antd.nist.gov/snap-all.html

NLnetLabs. *A short history of DNSSEC*. Internet, September 2012a. Last Accessed on 24 October 2012.
URL http://nlnetlabs.nl/projects/dnssec/history.html

NLnetLabs. *Dnssec-Trigger*. Internet, 2012b. Last Accessed on 24 October 2012.
URL http://www.nlnetlabs.nl/projects/dnssec-trigger/

OHMO. *Status map of DNSSEC deployment in ccTLD and gTLD*. Internet, November 2012. Last Accessed on 24 October 2012.
URL http://www.ohmo.to/dnssec/maps/

Ollmann, G. *The Pharming Guide (part 2)*. Online, 2007. Last Accessed on 24 October 2012.
URL http://www.technicalinfo.net/papers/Pharming2.html

Osterweil, E., Ryan, M., Massey, D., and Zhang, L. *Quantifying the operational status of the DNSSEC deployment*. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 231–242. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-334-1. doi:10.1145/1452520.1452548. Last Accessed on 24 October 2012.

Osterweil, E. and Zhang, L. *Securing the Domain Name System :Interadministrative Challenges in Managing DNSKEYs*. In *Security and Privacy Magazine*. 2009. Last Accessed on 24 October 2012.
URL http://irl.cs.ucla.edu/papers/j5ost.pdf

Perdisci, R., Antonakakis, M., Luo, X., and Lee, W. *WSEC DNS: Protecting Recursive DNS Resolvers from Poisoning Attacks*. In *IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN '09.*, pages 3–12. 2009. Last Accessed on 24 October 2012.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.188.3896&rep=rep1&type=pdf

Pokai Chen, S. S. *Security for Future Internet Architecture - Motivation from DNSSEC*. *Reliability Society 2010 Annual Technical Report*, 2010. Last Accessed on 24 October 2012.
URL http://paris.utdallas.edu/IEEE-RS-ATR/document/2010/5-Security%20for%20Future%20Internet%20Architecture_Shieh.pdf

PowerDNS. *Total number of DNSSEC delegations in .NL zone*. Internet, December 2012. Last Accessed on 24 October 2012.
URL https://xs.powerdns.com/dnssec-nl-graph/

Rasmussen, R. *DNSSEC Deployment in .GOV: Progress and Issues*. *2011 OARC Fall Workshop*, 2011. Last Accessed on 24 October 2012.

URL https://www.dns-oarc.net/files/workshop-201110/GOV%20DNSSEC%20Review.pdf

Roberts, L. *The Arpanet and computer networks.* In *Proceedings of the ACM Conference on The history of personal workstations*, HPW '86, pages 51–58. ACM, New York, NY, USA, 1986. ISBN 0-89791-176-8. doi:10.1145/12178.12182. Last Accessed on 24 October 2012.

Santcroos, M. and Kolkman, O. M. *DNS Threat Analysis. Mnet labs*, May 2007. Last Accessed on 24 October 2012.
URL http://nlnetlabs.nl/downloads/se-consult.pdf

Schlyter, J. and Griffin, W. *IETF RFC 4255. Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints*, January 2006. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc4255.txt

Schneier, B. *Applied Cryptography.* ISBN 0-471-12845-7. Katherine Schowalter, 1996. Last Accessed on 24 October 2012.

SecSpider. *SecSpider the DNSSEC Monitoring Project.* Internet, August 2012. Last Accesses on 30/07/2012.
URL http://secspider.cs.ucla.edu/

Seshadri, S. *DNSSEC in Windows. 33rd ICANN Meeting Cairo, Egypt*, 2008. Last Accessed on 24 October 2012.
URL http://cai.icann.org/files/meetings/cairo2008/seshadri-dnssec-windows-05nov08.pdf

Son, S. and Shmatikov, V. *The Hitchhiker's Guide to DNS Cache Poisoning.* In *SECURECOMM'10*, pages 466–483. 2010. Last Accessed on 24 October 2012.

Stoyanov, V. *DNSSEC .GOV Uptake. FOSE 2012.Making Dnssec the trust Infrastructure:Where Domain Name Security Is Headed.*, 2012. Last Accessed on 24 October 2012.
URL https://www.dnssec-deployment.org/wp-content/uploads/2012/03/2-Stoyanov-GOV-Uptake.pptx

Su, Z.-S. *IETF RFC 830. A Distributed System for Internet Name Service*, 1982. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/rfc/rfc830.txt

Su, Z.-S. and Postel, J. *IETF RFC 819. The Domain Naming Convention for Internet User Applications*, 1982. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/rfc/rfc819.txt

Timmers, M. *Research Project 1 - DNSCurve Analysis.* 2009. Last Accessed on 24 October 2012.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.5200&rep=rep1&type=pdf

Tzur-David, S., Lashchiver, K., Dolev, D., and Anker, T. *Delay Fast Packets (DFP): Prevention of DNS Cache Poisoning.* In *SecureComm*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 303–318. Springer, 2009. Last Accessed on 12 February 2013.
URL http://w3.cs.huji.ac.il/~dolev/pubs/dfp-selfcopy.pdf

van Rijswijk-Deij, R. *Deploying DNSSEC: Validation on recursive caching name servers.* 2012. Last Accessed on 24 October 2012.
URL http://www.surfnet.nl/Documents/rapport_Deploying_DNSSEC_v20.pdf

Vixie, P. *IETF RFC 2671. Extension Mechanisms for DNS (EDNS0)*, 1999. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc2671.txt

Vixie, P. *Whither DNSCurve? ISC Blog*, 2010. Last Accessed on 24 October 2012.
URL http://www.isc.org/community/blog/201002/whither-dnscurve

Vixie, P., Gudmundsson, O., and Wellington, B. *IETF RFC 2845. Secret Key Transaction Authentication for DNS (TSIG)*, May 2000. Last Accessed on 12 February 2013.
URL http://www.ietf.org/rfc/rfc2845.txt

Weiler, S. *IETF RFC 5074. DNSSEC Lookaside Validation (DLV)*, 2007. Last Accessed on 12 February 2013.
URL http://tools.ietf.org/html/rfc5074

Whaley, B., nemeth, E., Snyder, G., and Hein, T. R. *UNIX and Linux Administration Handibook.* ISBN: 0-13-148005-7. Pearson Education, 2010.

Williams, M. *AT&T Hit by DDoS Attack, Suffers DNS Outage.* Internet, August 2012. Last Accessed on 24 October 2012.

URL http://www.pcworld.com/article/260940/atandt_hit_by_ddos_attack_suffers_dns_outage.html

Yang, H., Osterweil, E., Massey, D., Lu, S., and Zhang, L. *Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC.* *IEEE Trans. Dependable Secur. Comput.*, 8(5):656–669, September 2011. ISSN 1545-5971. doi: 10.1109/TDSC.2010.10. Last Accessed on 24 October 2012.
URL http://dx.doi.org/10.1109/TDSC.2010.10

York, D. *Challenges and Opportunities In Deploying DNSSEC A progress report on an investigation into DNSSEC deployment.* Securing and Trusting internet Names, SATIN 2012. 2012. Last Accessed on 24 October 2012.
URL http://conferences.npl.co.uk/satin/papers/satin2012-York.pdf