Securing DNS

What consequences do the differences in design of DNSCurve and DNSSEC have on the implementations?

Author

Cornel de Jong
Cornel.dejong@os3.nl

26-06-2009

Supervisors

Fred Mobach (Systemhouse Mobach BV)
Mendel Mobach (Systemhouse Mobach BV)

Cees de Laat (UvA)

UNIVERSITEIT VAN AMSTERDAM

System and Network Engineering
Universiteit van Amsterdam
Spui 21
1012WX Amsterdam

Abstract

The Domain Name System (DNS) is a key infrastructure component of the Internet architecture. The fact that data can be manipulated to serve a malicious purpose has never been taken into consideration when the DNS was first developed. This can result in an untrustworthy DNS, last year D. Kaminsky showed the potential impact of this problem once again. Many solutions to secure the DNS focus on the protection of the name servers instead of the DNS as a whole.

This report shows the differences between DNSCurve and DNSSEC, two techniques developed to secure the DNS using a very different approach. DNSCurve offers authentication and encryption to the link-layer whereas DNSSEC offers message authentication and integrity verification through cryptographic signatures. The report is based on theoretical research to investigate the differences between these two techniques and will cover multiple areas, like: Installation requirements, transport protocol, challenges and tools. The maturity of DNSSEC, the existing signed ccTLDs, multiple testbeds and ongoing development make it more reliable for now than DNSCurve does. During the research period the ICANN announced steps to sign the root zone by the end of 2009; this is a major improvement for DNSSEC deployment. DNSCurve also shows some movement, the website now states that the software is under development and testing at the moment of writing (June 2009). DNSCurve looks very promising but first have to prove itself.

Acknowledgement

I would like to express my gratitude to Fred Mobach and Mendel Mobach for their information, guidance and advice during my research. Special thanks goes out to Thijs Stuurman for his review and comments.

Preface

This report has been written as part of my Master of Science study in System and Network Engineering at the University of Amsterdam. I have done research on securing the Domain Name System through two techniques: DNSCurve and DNSSEC and in particular the implementation differences between them. The research was performed under supervision of Fred and Mendel Mobach from Systemhouse Mobach BV.

Contents

1. Introduction

The Domain Name System (DNS) as defined in RFC 1034 and RFC 1035 is a key infrastructure component of the Internet architecture as we know it. It is the standard mechanism for domain name to Internet Protocol (IP) address translation and uses one of the largest hierarchical distributed databases to accomplish that task.

The DNS protocol was designed to ensure availability and scalability, more than 20 years later we can conclude that it worked out pretty well. The fact that data can be manipulated to serve a malicious purpose has never been taken into consideration when the DNS was first developed. Throughout the past 15 years steps have been taken to add an element of security and trustworthiness to the DNS. In 1990 S. M. Bellovin discovered a critical security flaw in the DNS protocol which could easily be exploited. The paper in which he described his discoveries was withheld from publication for over four years because it described a serious vulnerability for which there was no feasible fix. In 1997, RFC 2065 first specifies DNS Security Extensions (DNSSEC) as a manner to add security and data integrity checking. This RFC was revised in 1999 and led to RFC 2535; unfortunately this RFC had its own problems which more or less prevented implementation. In 2004, RFC 3833 was written which holds a threat analysis of the DNS. Further work resulted in 2005 in DNSSEC-bis which is specified in RFC 4033 through 4035. Recent developments keep showing how there is a definite importance that steps will be undertaken towards a secure DNS infrastructure.

This report describes the research on two techniques that improve the security and integrity of the DNS, both using different approaches. i) The DNSCurve project that uses link-level public-key protection to secure DNS packets, developed by D. J. Bernstein but is not formal specified. ii) The DNSSEC-bis protocol as defined in RFC 4033 through 4035 that uses public-key signatures to secure DNS records specified by R. Austein (Internet Systems Consortium Inc) et al.

Despite the fact that there is no formal specification of DNSCurve nor production ready software implementations, the publicly available documentation has been used as a source for this research.
During the research period news came that the Internet Corporation for Assigned Names and Numbers (ICANN) will work with the U.S. Department of Commerce's National Telecommunications and Information Administration (NTIA), the National Institute of Standards and Technology (NIST) and VeriSign on the goal of an operationally Signed Root Zone as soon as feasible in 2009. A signed root clearly shifts advantages towards DNSSEC; however the focus of this research is of a theoretical kind which focuses on the implementation differences of these techniques.

1.1 Research question

The first week of the project consisted of preliminary research and resulted in the following research question:

> " *What consequences do the differences in design of DNSCurve and DNSSEC have on the implementations?* "

To answer this question, several sub questions were defined:

- Are there special hardware and software requirements?
- What are the transport protocol requirements?
- Are the Confidentiality, Integrity and Availability (CIA) well-protected?
- Which cryptographic algorithms are used and can they be modified?
- Are there key revocation scheme's available?
- What is the overhead of each technique?
- Are the techniques mature enough for implementation in production environments?
- Are there provisioning and monitoring tools available?
- Are there other techniques for securing the DNS?

1.2 Scope

The focus of this research project will be to look at the implementational differences of DNSCurve and DNSSEC from a theoretical perspective.

Despite the fact that there is no formal specification of DNSCurve nor production ready software implementations, the publicly available documentation will be used as a source for this research. An experiment setup will not be part of the research project (this includes no source code analysis and comparison).

Other techniques for securing the DNS will be introduced shortly as they are less important to this research. The cryptographic algorithms will also be handled shortly because it requires sophisticated knowledge of elliptic curve cryptography.

1.3 Outline

Chapter 1 describes the research question and the scope of this research, chapter 2 continues with background information on the DNS and why protection is needed. Chapter 3 and 4 give an overview of DNSCurve and DNSSEC. Hardware and software requirements can be found in chapter 5, chapter 6 and 7 continue with the transport layer protocols and the cryptographic algorithms that are used. The challenges that an implementation faces are discussed in chapter 8. Tools to support the DNS administrator are discussed in chapter 9. Until the root zone is signed interim solutions can provide additional security, some of these techniques are discussed in chapter 10. Chapter 11 gives the conclusions on the research question and chapter 12 closes the report with some topics that might be worth future research.

2. Background

This chapter discusses the reason for this project and explain why it is necessary to secure the existing Domain Name System against malicious purposes.

2.1 Domain Name System

The Domain Name System (DNS) as defined in RFC 1034[1] and RFC 1035[2] is a key infrastructure component of the Internet architecture as we know it. It is the standard mechanism for domain name to Internet Protocol (IP) address translation and uses one of the largest hierarchical distributed databases to accomplish that task. The basic principle of the DNS is the use of human-friendly names for internet addresses, as names are human-readable and memorable, instead of numbers (IP addresses), which are actual only practical for computers.

There is also reverse DNS, which is the mechanism for Internet Protocol (IP) address to domain name translation. The reverse DNS database of the Internet is located in the Address and Routing Parameter Area (ARPA) Top Level Domain (TLD) of the Internet. IPv4 uses the in-addr.arpa[3] domain and IPv6 uses the ip6.arpa domain. Reverse resolving of an IP address is facilitated with the pointer (PTR) DNS record.

All the DNS data is stored in data structures called Resource Records (RRs) and each RR has an associated name, class and type. For example, an IPv4 address for www.example.com is stored in a RR with name www.example.com, class IN (Internet) and type A (IPv4 address). A multihomed host with multiple IPv4 addresses has several RRs, with the same name, class and type but with different IPv4 addresses. The set of all Resource Records that share the same name, class and type is called a Resource Record Set (RRset).

As mentioned before, the DNS is a distributed database organized in a tree structure. The top of the tree represents the root zone, which delegates authority to TLDs (for example .com .net .org .gov .mil). The .net zone delegates authority to create dnssec.net., .org delegates authority to create iana.org. etc. This results in a DNS tree structure where every node corresponds to a zone and every zone belongs to a single administrative authority. Which is served by multiple authoritative name servers that provide name resolution for all the names in the zone. Each RRset in the DNS belongs to a specific zone and is stored at the name server of that zone.

2.1.1 DNS components

In essence the DNS can be divided into three main components. These components can be used to define all kinds of subtleties, for example name servers: authoritative, master, primary, slave, secondary, recursive, etc. all exist. Their functions may slightly differ but in essence it are all name servers. The three main components described below are cited from RFC 1034 and a graphical representation can be found in figure 1.

- The Domain Name Space and Resource Records
- Name Servers
- Resolvers

The DOMAIN NAME SPACE and RESOURCE RECORDS, which are specifications for a tree structured name space and data associated with the names. Conceptually, each node and leaf of the domain name space tree names a set of information, and query operations are attempts to extract specific types of information from a particular set. A query names the domain name of interest and describes the type of resource information that is desired.

NAME SERVERS are server programs which hold information about the domain tree's structure and set information. A name server may cache structure or set information about any part of the domain tree, but in general a particular name server has complete information about a subset of the domain space, and pointers to other name servers that can be used to lead to information from any part of the

domain tree. Name servers know the parts of the domain tree for which they have complete information; a name server is said to be an AUTHORITY for these parts of the name space. Authoritative information is organized into units called ZONEs, and these zones can be automatically distributed to the name servers which provide redundant service for the data in a zone.

RESOLVERS are programs that extract information from name servers in response to client requests. Resolvers must be able to access at least one name server and use that name server's information to answer a query directly, or pursue the query using referrals to other name servers. A resolver will typically be a system routine that is directly accessible to user programs; hence no protocol is necessary between the resolver and the user program.

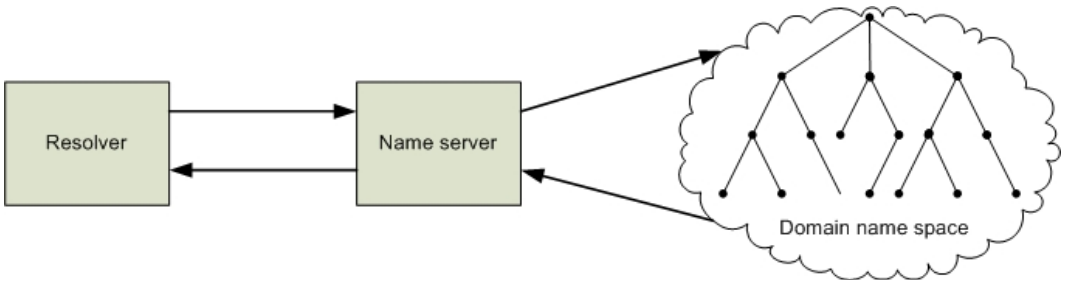


Figure 1. A graphical representation of the Domain Name System.

Figure 2 is a graphical representation of the before mentioned domain name space and gives a simplified overview. Here we can identify the Resource Records, different zones and delegation.

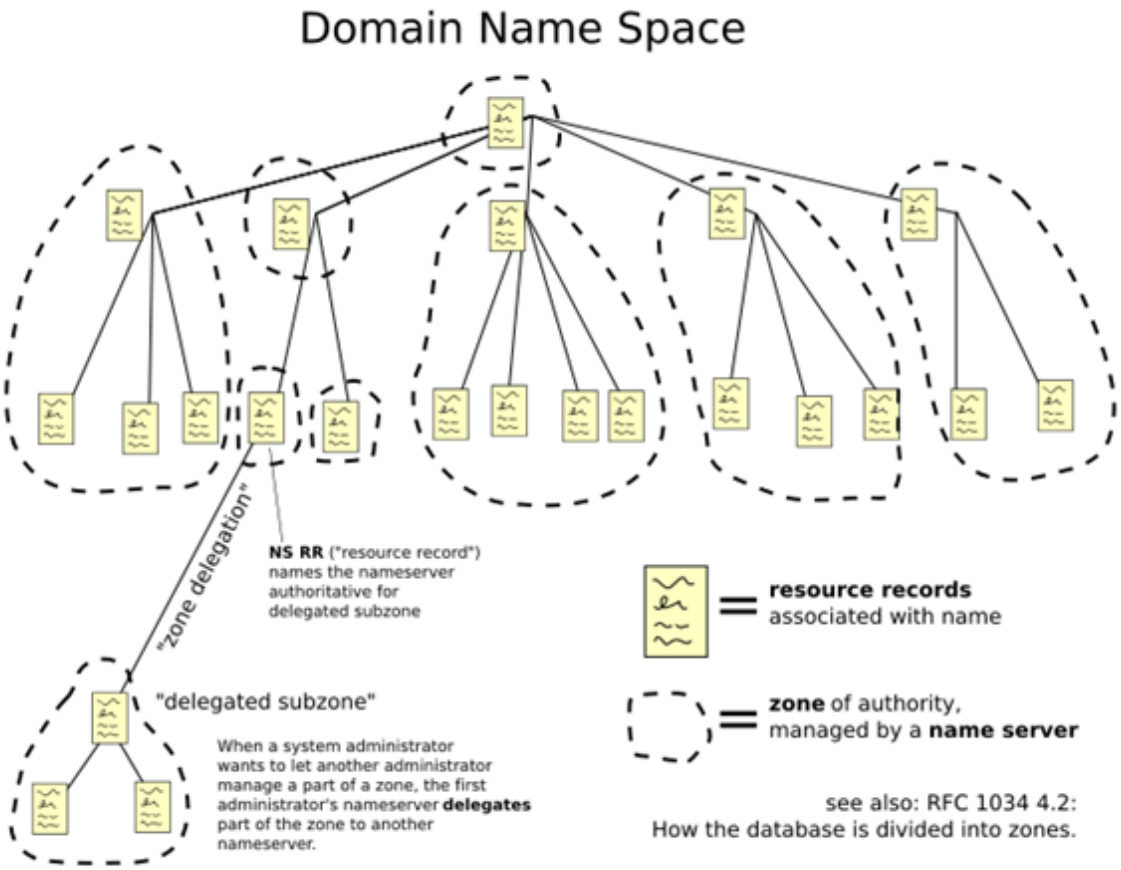


Figure 2. The Domain Name Space. Courtesy of Wikipedia.

2.1.2 Domain names

Domain names are expressed as a sequence of labels, separated by a period "." and ending with the (empty) root label. The empty root label is represented as a single period ".". Domain names can be distinguished in Fully Qualified Domain Name[4] (FQDN) or Partially Qualified Domain Name (PQDN).
A FQDN defines the complete domain name that uniquely identifies a node in the DNS Name Space. It does this by giving the full path of labels from the root down to the node and thereby defines the absolute location of a domain. The opposite is a PQDN which defines a portion of a domain name. This is a relative name that has only a meaning within a particular context. So multiple similar PQDNs can exist while they have no unique meaning.

2.1.3 Resource Records

Besides the earlier mentioned "A" Resource Record (RR) there are many other RR types, the most important[5] ones are mentioned in table 1 below. Note that this is not an exhaustive collection. DNSCurve uses the NS RR and DNSSEC introduces new RR that will be covered later on.

Record Type	Associated entity	Description
SOA	Zone	Start Of Authority. Contains information about the represented zone
A	Host	Address. Contains an IP address associated with this host
MX	Domain	Mail exchanger. Refers to a mail server and specifies a priority
SRV	Domain	Server selection. Refers to a server handling a specific service
NS	Zone	Name server. Refers to a name server for the represented zone
PTR	Node	Pointer. IP address to name
CNAME	Host	Canonical name. Canonical name of a host
HINFO	Host	Host information. Contains information about the host it represents
TXT	Any kind	Text. Contains any kind of useful information

Table 1. Most common Resource Record types.

2.2 DNS evolution

The DNS is known to be susceptible to malicious purposes; this reduces the trustworthiness of the system. In 1990 S. M. Bellovin discovered a critical security flaw in the DNS protocol which could easily be exploited. The paper[6] in which he described his discoveries was withheld from publication for over four years because it described a serious vulnerability for which there was no feasible fix. The citation below is derived from his paper:

> "*As we have stated before, reliance on host addresses or host names for authentication is fundamentally flawed. The only real security in an internetworking environment is cryptographic. The Kerberos system is probably the best choice today; though flawed in places, it is far better than the current scheme.*"

It is interesting to see that almost two decades ago S. M. Bellovin already said that the only real security with internetworking lies in using cryptography. On the other hand we see that cryptographic algorithms show flaws, again this is what we see today with for example: DES, MD5, RSA with certain key lengths, etc. His discovery started the discussion about security in the DNS. RFC 3833[7] states that the first discussions go back to 1993:

> "*The earliest organized work on DNSSEC within the IETF was an open design team meeting organized by members of the DNS working group in November 1993 at the 28th IETF meeting in Houston. The broad outlines of DNSSEC as we know it today are already clear in Jim Galvin's summary of the results of that meeting* "

Finally, in 1997 the first DNS Security Extensions (DNSSEC) RFC was specified: RFC 2065[8]. This was the first attempt to add security and data integrity checking to the DNS. Problems with implementing this RFC led to RFC 2535[9], a revision of the original specification. Unfortunately this RFC had its own problems (especially scaling problems) which prevented implementation at a large scale.

While the first two DNSSEC RFCs were developed, the Internet Engineering Task Force (IETF) had never specified the specific set of threats against which DNSSEC is designed to protect. This finally happened in RFC 3833 in 2004.

To solve the problems introduced by earlier RFCs, the IETF modified DNSSEC and called it DNSSEC-bis which was formalized in RFCs in RFC 4033[10] [11] [12] through 4035 in 2005. These RFCs turned out to have their own security issues, the most important one is zone enumeration (aka zone walking). The Next Secure (NSEC) record made it possible to retrieve the entire list of names or other information in a zone. This finally got solved by "DNSSEC Hashed Authenticated Denial of Existence" (NSEC3) in RFC 5155[13] in 2008. Table 2 gives an overview of the most important RFCs in the DNS evolution.

Another observation is that the implementation of DNSSEC makes it easier to conduct a Denial of Service (DoS) attack, due to the larger response messages. It should be noticed that almost every DNSSEC RFC contains crucial problems and often gets updated by a new RFC.

RFC	Description	Year
Original DNS		
882	Domain Names - Concepts and Facilities	November 1983
1034 1035	Domain Names - Concepts and Facilities Domain Names - Implementations And Specification	November 1987
DNSSEC		
2065	Domain Name System Security Extensions	January 1997
2535	Domain Name System Security Extensions	March 1999
Extensions and Threat analysis		
2671	Extension Mechanisms for DNS (EDNS0)	August 1999
3833	Threat Analysis of the Domain Name System (DNS)	August 2004
DNSSEC-bis		
4033 4034 4035	DNS Security Introduction and Requirements Resource Records for the DNS Security Extensions Protocol Modifications for the DNS Security Extensions	March 2005
5155	DNS Security (DNSSEC) Hashed Authenticated Denial of Existence	February 2008

Table 2. Overview of the DNS evolution.

2.3 Threat analysis of the DNS

The RFC that contains the threat analysis of the Domain Name System was written in 2004 and describes the specific set of threats against which DNSSEC is designed to protect. Appendix 2 shows the main threats against DNS in a graphical representation. The attacks are organized into different categories as mentioned below. Some of them have a more common name added to it.

- Packet interception: Man-In-The-Middle attacks
- ID guessing and query prediction
- Name chaining: Cache poisoning
- Betrayal by trusted server
- Denial-of-Service
- Wildcards insertion

Packet interception can be hard to prevent, especially when someone is in-line with the traffic. Wireless networks making it easier to conduct this attack. ID guessing and query prediction are often used together with cache poisoning. With the discovery of the Kaminsky bug[14] last year, query prediction got an enormous speed boost. This resulted in better randomization in name servers to prevent these attacks.

Betrayal by a trusted server can happen for example with customers using the DNS server from their Internet Service Provider (ISP). They blindly trust their ISPs DNS and the responses that get back, this can also be used for censorship. Denial of Service is becoming even easier with DNSSEC, send a tiny query with a forged origin address and get a huge response back. If this is done from a botnet it is even harder to trace the real origin.

It can be difficult to provide integrity for wildcard DNS records as they will match queries for non-existent domain names in a zone.

2.4 CIA Triangle

The CIA Triangle describes Confidentiality, Integrity and Availability. These aspects must be in balance, and are often represented by a triangle. The section below describes the elementary weaknesses in DNS and how DNSCurve and DNSSEC handle this.

Confidentiality
Confidentiality is concerned with the prevention of disclosure of information to unauthorized individuals or systems.

Confidentiality has never been part of the design of DNS or DNSSEC. The information is meant to be public. However the zone files are often considered as confidential, because they provide a blueprint of the network architecture. DNSCurve encrypts all DNS packets providing some confidentiality.

Integrity
Integrity is based on the trustworthiness of information resources and means that the data cannot be created, modified or deleted without authorization.

DNSCurve cryptographically authenticates all DNS responses at the link-layer whereas DNSSEC provides signatures to RRsets, making it easy to verify if messages are modified between signing and verifying. The integrity can be compromised trough cache poisoning or betrayal of trusted servers, without DNSCurve or DNSSEC it is extremely difficult to verify the integrity of the received messages.

Availability
Information must be available when needed by authorized users. When an information system is not available when needed it is nearly as bad as no system at all.

The DNS has no protection against Denial of Service (DoS) attacks. The use of User Datagram Protocol (UDP) makes it easy to initiate such an attack whereas using Transmission Control Protocol

would use more resources. DNSCurve claims to quickly recognize and discards forged packets. DNSSEC can do little about DoS attacks and one could argue it increases the risk of a DoS. Sending a tiny query can result in a massive response.

3. DNSCurve

DNSCurve offers authentication and encryption to the link-layer, using elliptic curve cryptography. The recently updated (2009-06-22) DNSCurve website[15] is now showing that DNSCurve is part of a larger project to encrypt and authenticate all Internet packets and that the techniques used in DNSCurve are easily adapted to other Internet protocols.

DNSCurve has been co-developed with a public-domain Networking and Cryptography library[16] (NaCl). NaCl (pronounced "salt") is a new easy-to-use high-speed software library for network communication, encryption, decryption, signatures, etc. The goal of NaCl is to provide all of the core operations needed to build higher-level cryptographic tools. NaCl is part of the Computer Aided Cryptography Engineering[17] (CACE) project. The goal of CACE is to develop a toolbox that supports the production of high quality cryptographic software. The CACE project is co-financed by the European Commission's Seventh Framework Programme[18] (FP7).

The information regarding DNSCurve is originating from the official website. However no independent benchmarks are available nor a formal specification of DNSCurve. For its cryptographic functions it uses elliptic curve cryptography with a specific curve: Curve25519 (modified by D. J. Bernstein).

Due to the lack of a formal specification, there is no official description other than on the website. The services provided by DNSCurve are shown below:

- DNSCurve encrypts all DNS packets, while the DNS sends this in an unencrypted form and even DNSSEC does not encrypt the packets. (Based on the design principles of the DNS this information is not encrypted.)
- DNSCurve cryptographically authenticates all DNS responses. This should be enough to eliminate forged DNS packets. In the DNS, queries and responses are secured by a Transaction ID (TXID), a UDP source-port randomization and perhaps some vendor specific solutions. This does not prevent forged DNS records, it only makes it harder.
- The DNS has no protection against Denial of Service (DoS) attacks. DNSCurve can recognize and discard forged DNS packets.

4. DNSSEC

DNSSEC offers[19] message authentication and integrity verification through cryptographic signatures. It can authenticate the DNS origin and integrity by verifying that no modifications between signing and validation took place. But it does not provide authorization nor confidentiality. Figure 3 gives an overview of the possible DNSSEC traffic flow.

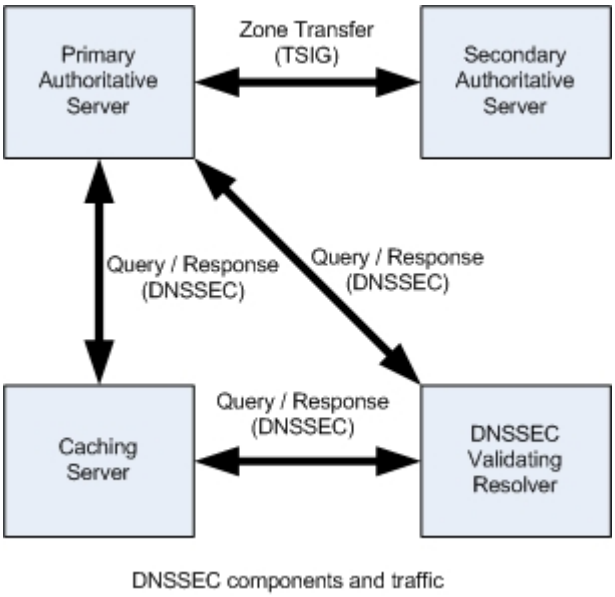


Figure 3. The DNSSEC components and traffic.

According to RFC 2535, the scope of DNSSEC can be categorized into three services (cited):

- Key distribution
- Data origin authentication
- Request and transaction authentication

Key distribution
A Resource Record (RR) format is designed to associate keys with DNS names. This makes it possible to use the DNS as a public key distribution mechanism in support of DNS security itself and other protocols

Data origin authentication
Authentication of the origin is provided by associating with RRsets in the DNS cryptographically generated digital signatures.

Request and transaction authentication
The data origin authentication service described above protects retrieved RRs and the non-existence of RRs but provides no protection for DNS requests or for message headers

In order to provide these services four new Resource Records were introduced:

- DNSKEY (DNS Public Key)
- RRSIG (Resource Record Signature)
- NSEC(3) (Next Secure)
- DS (Delegation Signer)

On the next page is a short description[20] of the four Resource Records, followed by an overview of the signing process in figure 4.

DNSKEY
A private and public key pair must be generated for each DNSSEC secured zone. The private key must remains secret and the public key is published in the zone file as a DNSKEY Resource Record (RR). The zone itself is signed using the private key of the key pair. DNSSEC can use the following algorithms for digital signatures:

- RSA/MD5 (NOT RECOMMENDED)
- DSA/SHA1 (OPTIONAL)
- RSA/SHA1 (MANDATORY)

From the algorithms above, only RSA/SHA1 should be used (as it is MANDATORY in the RFC) and with a key length of at least 1024-bit.

Zone signing uses two types of keys, the first is a Zone Signing Key (ZSK), and the second a Key Signing Key (KSK). The ZSK is used to sign the RRsets within the zone, including signing the ZSK itself. The KSK is used to sign the keys of the zone, including the ZSK and KSK.

RRSIG
An RRset is a set of RRs within a zone that share the same name, class and type. With DNSSEC, RRsets are digitally signed. The signature is based on a hash generated of the RRset and then encrypt the hash using the zone's private key (ZSK). The DNSSEC validating resolver can verify the integrity of the RRset by validating the digital signature using the public key.

NSEC(3)
After a zone is signed, a NSEC RR is added after each RR to create a chain of valid names in the zone file. The last NSEC RR in the chain points back to the zone apex or root. This creates the opportunity for an authenticated "Denial of existence" (no-such-record) in response to a query. This effectively maps the entire content of the zone file and making it possible through zone enumeration (a.k.a. zone walking) to reveal this information. So zone transfers might be prohibited, but zone enumeration makes it possible to retrieve the content of the zone file.

Many people considered this as a security flaw; NSEC3 was introduced as a solution. NSEC3 uses a hashing algorithm on the names in a zone and uses a hashed ordering of these names. This solution comes with a new NSEC3 RR. This solution does not solve the zone enumeration problem; it just makes it harder to reveal the information. An attacker could download the hashes and start brute forcing them offline. It is recommended to regularly chance the key and salt used for the hashing.

DS
A Delegation Signer RR can be used to add a secured zone to an existing chain-of-trust or to secure delegation to a sub-zone. The DS RR helps a client to validate the public key of a zone, the DNSKEY RR. This is done by using a chain-of-trust in the hierarchical delegation structure of the DNS. The DS RR contains a hash of the public key of the child zone. This record is signed using the private key of the parent zone with a matching RR. When validating a zone, the associated DS, RRSIG (DS) and DNSKEY of the parent zone are retrieved. The RRSIG (DS) record is decrypted by using the DNSKEY and can then be validated by checking that the result matches the DS record. This is the public key of the zone, according to the parent of the zone. This can be compared with the DNSKEY record of the zone; this relies on the parent zone key. This process continues when DNSSEC finds a trusted key, this should be the DNSKEY of the root zone. But the root is not signed, so validation would fail unless an interim solution like DNSSEC Lookaside Validation (DLV) or Interim Trust Anchor Repository (ITAR) is used.

DNSSEC-bis also introduces two new message header bits:

- CD (Checking Disabled)
- AD (Authenticated Data)

CD-bit
A DNSSEC name server will not perform signature validation for authoritative data while processing the query, even when the CD-bit is clear. A DNSSEC name server should clear the CD-bit when sending an authoritative response.

AD-bit
The AD-bit is set by a DNSSEC name server if all the RRsets in the Answer and Authority sections of the response are authentic.

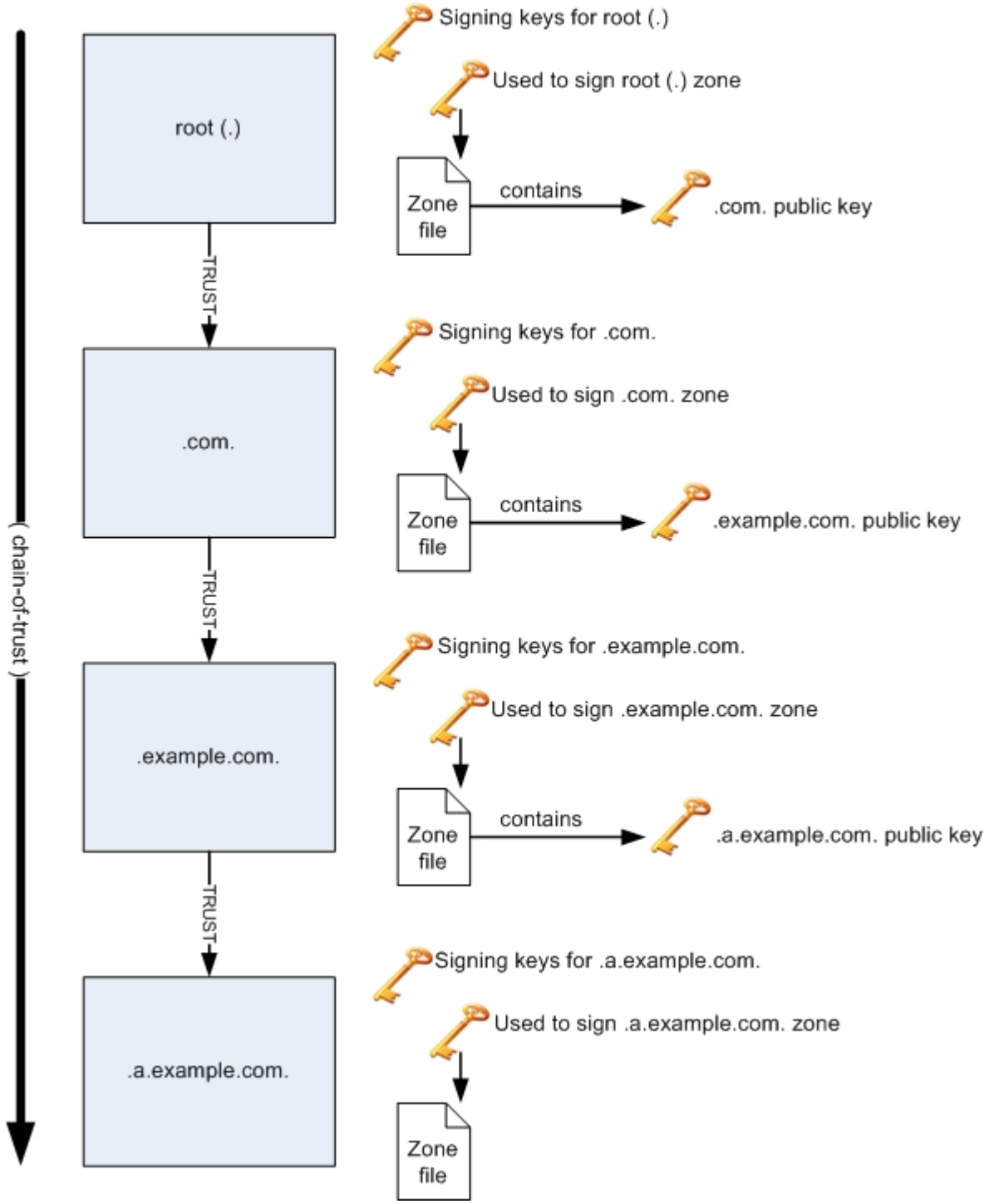


Figure 4. The DNSSEC Signing process.

5. Installation requirements

This chapter describes the hardware and software requirements that are necessary for implementing DNSCurve and DNSSEC. Requirements can depend on the size of the organization and the existing DNS infrastructure.

5.1 DNSCurve

Notice that the DNSCurve website has been regularly updated since 2009-06-22 and that some information has changed. An important line that has been added on the introduction page states that:

> "*DNSCurve is part of a larger project to encrypt and authenticate all Internet packets. The techniques used in DNSCurve are easily adopted to other Internet protocols.*"

This information was not available at the start of this research project and shows us that it is not only designed to protect the DNS. This might change the discussion about DNSCurve into a more general one "securing internet traffic".

There are two different DNSCurve enabled servers that can be installed:

- DNSCurve cache (recursive).
- DNSCurve forwarder (authoritative).
  DNSCurve forwarder (Stand-alone forwarder, prevents changes to the existing DNS server).

The stand-alone DNSCurve forwarder is in essence a normal DNSCurve forwarder.

5.1.1 DNSCurve cache

The website about the DNSCurve cache software, clearly states that there is no official released software available yet. The citation below shows the status at the moment of writing 2009-06-25:

> "*DNSCurve cache software is, at the time of this writing (June 2009), undergoing development and testing.*"

The installation looks pretty straight forward; simply upgrade your recursive server to one that supports DNSCurve (see citation below). However this requires the adoption of DNSCurve in DNS software, surely D. J. Bernstein's dnscache software will support it in the future. But it is assumable that other vendors will are not willing to adopt DNSCurve in the near future. It is more likely that they will (if not already) support DNSSEC and perhaps some interim solutions to avoid cache poisoning.

> "*simply upgrade your DNS cache (a "recursive server" such as dnscache or PowerDNS Recursor or BIND or MaraDNS or Nominum CNS or Unbound) to a DNS cache that supports DNSCurve.*"

Based on the available information on the website, there is also stated that no additional cache configuration is required. Also when positioned behind a firewall no additional configuration is required. Although when using deep packet inspection on DNS packets this might introduce problems because DNS / DNSSEC traffic is still readable instead of the encrypted traffic in DNSCurve, however this is not very common.

5.1.2 DNSCurve forwarder

The website about the DNSCurve forwarder software, clearly states that there is no official released software available yet. The citation below shows the status at the moment of writing 2009-06-25:

> "*DNSCurve forwarder software is, at the time of this writing (June 2009), undergoing development and testing.*"

The installation of the DNSCurve forwarder can be done using two different approaches. It can be installed similar as the DNSCurve cache server, by installing an authoritative DNS server that supports DNSCurve. The other approach uses a dedicated DNSCurve forwarder, which does not require changes to the existing DNS server.

The installation of an authoritative DNS server supporting DNSCurve faces the same hurdles as the recursive server; it requires the adoption of DNSCurve through DNS software vendors. Surely D. J. Bernstein's tinydns software will support it in the future. But again adoption depends on other vendors.

5.1.3 DNSCurve stand-alone forwarder

Perhaps a better way to get the DNSCurve adoption started without requiring updating or changing the existing DNS server software is to use the stand-alone DNSCurve stand-alone forwarder. The website describes the five steps that are necessary to achieve the installation (cited):

1. Install the DNSCurve forwarder on a new UP address. (If you install the forwarder on the same computer as your existing DNS server then you need to put it on a different IP address from the existing DNS server.
2. Configure the DNSCurve forwarder to forward to your existing DNS server's IP address.
3. Add, in your DNS data, a special DNSCurve server name for the DNSCurve forwarder. The name is specific to this DNSCurve forwarder and is automatically generated during installation of the forwarder.
4. Add the same DNSCurve server name in your parent DNS data.
5. After a week, remove the old non-DNSCurve server names.

According to the DNSCurve website, that is enough to get DNSCurve running. There is no need to change any other DNS data or procedures for updating DNS data.

5.1.4 Overview

Based on the available information there are no or minimal additional hardware requirements. Software needs to support DNSCurve, at the moment of writing there is no software supporting it. Probably the first DNS software supporting DNSCurve comes from D. J. Bernstein. The stand-alone forwarder software is available[21] for some time, however this is not official released. Research[22] by M. Timmers shows that the software is not mature and requires changes in the source code to get it working.

5.2 DNSSEC

The requirements for DNSSEC are somewhat more complicated. But can be categorized into:

- **Hardware**, new or existing. This depends on the size of the organizations and the frequency by which the zone file changes. Normal organizations have relatively static zone files whereas Registries and Registrars have constantly updated zone files.

- **Name servers**, the DNS name servers must be capable of DNSSEC, if not they should be updated or replaced.

- **Resolvers**, when only the name servers offer DNSSEC Resource Records (RRs), it will not help much. Resolvers must be capable of validating these RRs, this also requires computing power.

- **Key length**, the key length (1024, 2048, etc) configured with RSA/SHA1 has significant influence on the computing power required.

- **Re-engineering of the DNS infrastructure**, it might be necessary to change the existing DNS infrastructure to enable DNSSEC. This can also be used to clean the zone(s) from old RRs.

- **Tools**, tools are not necessarily a requirement for the implementation of DNSSEC but they can help DNS administrators.

There are several vendors offering DNSSEC capable name servers, for example: BIND, NSD and Unbound. Others offer partly support, for example: Microsoft Windows 2003 Server or PowerDNS. This should be carefully considered when choosing the DNSSEC software vendor. The different vendors may have their own requirements, but that lies outside the scope of this research. The ICANN did a survey[23] of DNSSEC Capable DNS Implementations.

5.2.1 Impact

One of the major obstacles to the implementation of DNSSEC is the concern about the performance impact DNSSEC will have on the existing infrastructure. Both name servers and resolvers need to do more work. The KSK and ZSK key generation and the signing of the RRs is done on a monthly and annual basis and using pre-computation (except when using dynamic updates). This increases the necessary computing resources for a limited amount of time (dynamic updates require some on-the-fly signing). Authoritative servers will be sending larger responses (also for non-existing RRs). The DNSSEC validating resolvers must perform the signature verification, which is also computationally expensive work.

More about the impact of DNS can be found on the website from the NIST[24], RIPE-NCC[25] and in this paper[26] from A. Guillard.

5.2.2 end-to-end validation

In order to provide end-to-end validation from the root to the customer it is necessary that Operating Systems (OSs) are capable of validating the DNSSEC information. But here comes the chicken-egg problem in place, DNSSEC is not implemented widely and customers do not ask for DNSSEC validation. Instead they often rely on the DNS server provided by their Internet Service Provider (ISP). Here lies a major change for the adoption of DNSSEC, when OS vendors start adding DNSSEC validating resolvers to their operating systems.

But there are some issues: what to do when validation fails?

- **Silently prevent the connection from being established.** This might annoy customers who can reach the domain from another non DNSSEC validating resolver.
- **An informational screen with the problem.** Most of the customers do not understand the error messages and just want the information they are looking for.
- **SSL like STOP sign.** Just like with SSL certificates, people might just click around it. Not understanding the importance of the messages and just want the information their looking for.

Within organizations, there is often a central DNS server that handles all the queries and responses. When OSs come with DNSSEC validating resolvers, will they be used in organizations? Then the validation happens twice, once at the central DNS server and again at the local computer. More interesting is how to react when validation fails. Organizations might want to use a corporate error page depending on the error. Such an error page can be very basic like: “A validation error has occurred, please inform your system administrator”, these messages should be logged and monitored.

Maybe there should be an emergency mode where DNSSEC validation can be turned off, in the case of a domain that is critical for the business becomes unavailable. Though this would offer security for functionality.

5.2.3 Windows and DNSSEC

Within organizations Microsoft Windows is commonly used as workstation operating system and also many home computers use it. So it is interesting to look at the DNSSEC validating resolver support it offers.

Windows XP
The resolver will cache the DNSSEC Resource Records (RRs) in same manner as any other RRs. But it does not perform any cryptography, authentication or verification. On the TechNet[27] website we read the following:

> "*The DNS client does not read and store a key for the trusted zone and, consequently, it does not perform any cryptography, authentication, or verification. When a resolver initiates a DNS query and the response contains DNSSEC resource records, programs running on the DNS client will return these records and cache them in the same manner as any other resource records. This is the extent to which Windows XP DNS clients support DNSSEC. When the DNS client receives the SIG RR relating to the RRset, it will not perform an additional query to obtain the associated KEY record or any other DNSSEC records."*

Windows Vista
Search results on the Microsoft website did not return information about Vista's DNSSEC capabilities.

Windows 7
Microsoft Windows 7 is the latest version and shows many security improvements. It is the first client operating system that can verify that the communication with a DNS server is secure and verify that the server performed DNSSEC validation on its behalf. This is still being tested, but it seems like, the actual validation takes place at the server (details about this are not available at this moment).

On TechNet[28] we read the following:

> "*Windows 7 is the first client operating system to include the necessary pieces to allow the client to verify that it is communicating securely with a DNS server and verify that the server has performed DNSSEC validation on its behalf. This technology is currently being tested to ensure the maximum compatibility with current Internet infrastructure and aims to play a continuing role in securing DNS data in the future.*"

6. Transport layer

Accessing name servers on the Internet can be done using UDP on port 53 or using TCP also on port 53. Actually all DNS traffic can use TCP but this is not preferred due to the overhead it introduces and the lower performance. But TCP is being used for specific functions like zone transfers. This chapter describes the advantages and disadvantages of using UDP or TCP.

6.1 Limitations

Originally the UDP message size is limited to 512 bytes (excluding the IP or UDP headers) in RFC 1035. Besides the message limitation there are some other size limitations:

- Labels 63 octets or less.
- Names 255 octets or less.
- TTL Positive values of a signed 32 bit number.

Messages that exceed this limitation are truncated and the TC-bit (Truncated) is set in the header. Queries send using UDP may get lost, because UDP is a stateless protocol it does not check if the message actually arrived. This requires a retransmission strategy. Because of the stateless protocol characteristics, queries and their responses may receive out of order, this means that resolvers should not depend on the order of receiving.

6.2 UDP versus TCP

UDP is a stateless protocol whereas the TCP protocol is stateful. A stateful protocol means that it needs to set up a session, keep track of the session and close the session, this requires more resources than a stateless protocol. A stateless protocol is more: “fire and forget”, it does not keep track whether or not messages are received. This makes TCP an expensive protocol for simple transactions like DNS. When TCP is used for al DNS traffic it would cause severe overhead and delays. Currently TCP is only used for specific tasks like: zone transfers and with DNSSEC when EDNS0 is not supported.

6.3 Consequences for DNSSEC

The limitations mentioned above, have consequences for DNSSEC. DNSEC Resource Records (RRs) can easily exceed the 512 bytes limitation, causing messages to be truncated. This limitation can also have consequences when DNS is using IPv6 addresses, as they use longer IP addresses, but this is not part of the research.

DNSSEC require the support of Extension Mechanisms for DNS (EDNS0), which is introduced in RFC 2671[29], also support for the DNSSEC OK (DO) EDNS header bit, described in RFC 3225[30] is required. This makes it possible for DNSSEC validating resolvers to indicate in its queries that it wishes to receive DNSSEC RRs in the response messages. EDNS0 makes it possible to support UDP message sizes up to 4096 bytes, however if EDNS0 is not supported fallback to TCP will happen, causing an increased load on the DNS server and increasing query latency.

6.4 EDNS Buffer sizes

To get more information about the EDNS buffer sizes currently in use. Statistics are needed; the figures below are originating from the f-root and the c-root servers. Appendix 1 shows the growing ENDS0 support and the differences in message sizes from 2006 till 2008.

Figure 5 shows the EDNS buffer sizes advertized by F root nodes in January 2006. Approximately 20% of the queries have a buffer size of 4096, another 20% of the queries have a buffer size of 2048, while on the other hand 55% does not support EDNS0. The rest, 5% uses 1024 as buffer size.

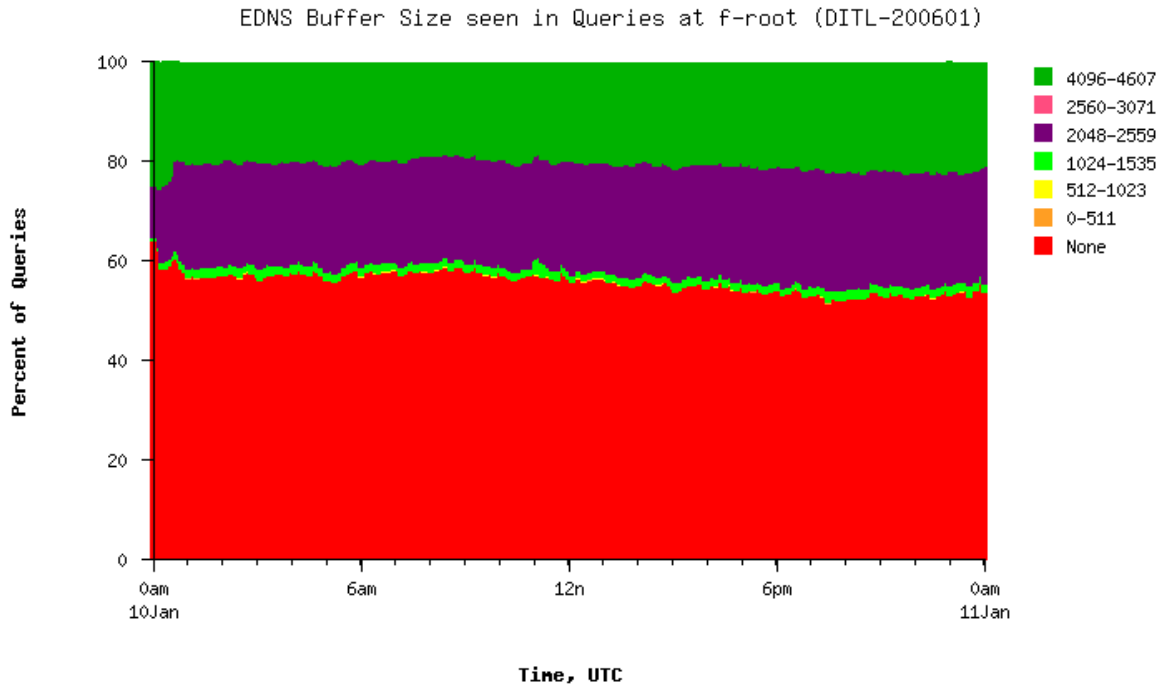


Figure 5. EDNS Buffer sizes at f.root-servers.net. Courtesy of DNS-OARC.

Figure 6 shows the EDNS buffer sizes advertized by C root nodes in June 2009. Approximately 55% of the queries have a buffer size of 4096, while on the other hand 40% still does not support EDNS0. The rest, 5% uses different buffer sizes.

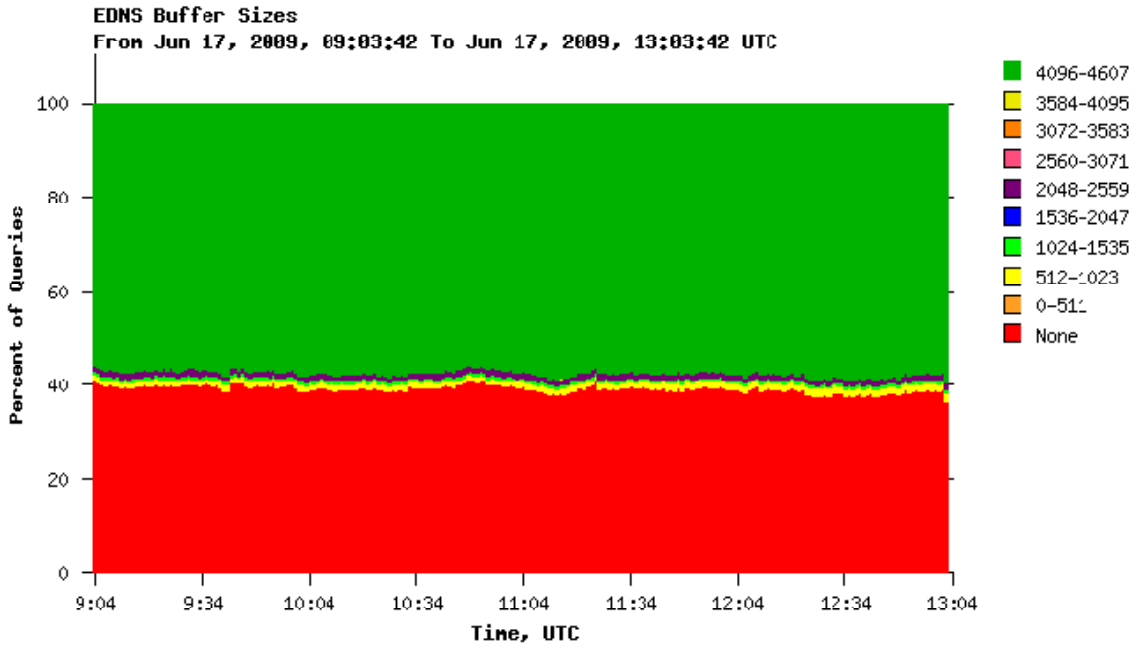


Figure 6. EDNS Buffer sizes at C.root-servers.net. Courtesy of D. Wessels and S. Castro.

6.5 Traffic after DNSSEC signing

It is important to know what will happen with the DNS traffic after a zone is signed with DNSSEC. For example: an increase in TCP traffic would demand more resources of the server and might result in upgrading or replacing a server.

Figure 7 shows the traffic of the .org zone before and after signing with DNSSEC. The amount of UDP queries stays the same whereas the amount of TCP queries makes a huge increase. This might be caused by name servers that do not support EDNS0; they will fall-back to TCP. Before the signing no truncated (TC-bit = 1) queries are seen, after the signing they show up. This can be explained due to the increased message size of DNSSEC. Interesting is the line showing EDNS with buffer size 512, why use that buffer size if the original DNS RFC already supports this size.

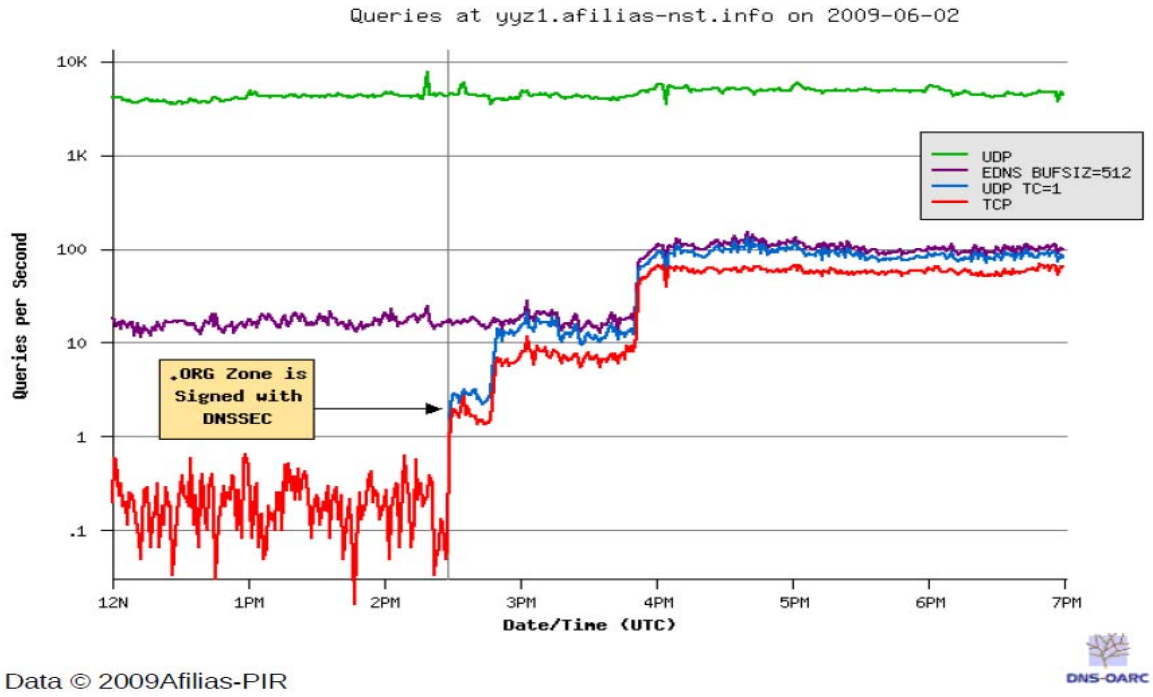


Figure 7. Traffic .org domain before and after signing with DNSSEC. Courtesy of Afilias-PIR.

7. Cryptographic Algorithms

One of the obvious differences between DNSCurve and DNSSEC is the cryptography algorithm that is used. DNSCurve uses elliptic curve cryptography (1985) and DNSSEC uses RSA (1977). The strength of the algorithms partly depends on the key length that is used; the key length however is part of discussion. It should be noticed that the key length cannot be compared between these two different cryptographic algorithms!

The two cryptographic algorithms will be discussed briefly.

According to RFC 4641[31]:

> "*Assuming this rich attacker will not attack your key and that the key is rolled over once a year, we come to the following recommendations about KSK sizes: 1024 bits for low-value domains, 1300 bits for medium-value domains, and 2048 bits for high-value domains.*
>
> *Whether a domain is of low, medium, or high value depends solely on the views of the zone owner. One could, for instance, view leaf nodes in the DNS as of low value, and top-level domains (TLDs) or the root zone of high value. The suggested key sizes should be safe for the next 5 years.*
>
> *As ZSKs can be rolled over more easily (and thus more often), the key sizes can be made smaller. But as said in the introduction of this paragraph, making the ZSKs' key sizes too small (in relation to the KSKs' sizes) doesn't make much sense. Try to limit the difference in size to about 100 bits.*"

According to the presentations[32] of D. J. Bernstein he suggests that:

> "*1024-bit RSA is irresponsible.*
> *2003: Shamir–Tromer et al. concluded that 1024-bit RSA was already breakable by large companies and botnets.*
> *2003: RSA Laboratories recommended a transition to 2048-bit keys "over the remainder of this decade."*
> *2007: NIST made the same recommendation.*"

Increasing the key length means more computing power is needed; depending on the size of the organization this can have serious consequences on the available resources. Following D. J. Berstein's suggestion all keys should at least be 2048-bit; no doubt it will be more secure. But will it be manageable on the existing hardware? Using 2048-bit keys in the leafs require all other keys in the chain to be at least that strong or it introduces a weaker link.

7.1 DNSCurve Cryptographic Algorithm

DNSCurve uses elliptic curve cryptography, in particular Curve25519. The description below is cited from the website to provide an overview of the working:

All DNSCurve communications are between two public keys. When a DNSCurve cache sends a packet to a DNSCurve server, it encrypts and authenticates the packet from its own public key to the server's public key. Similarly, when the server sends a packet back to the cache, it encrypts and authenticates the packet from the server's public key to the cache's public key.

Specifically, let's say the cache's long-term secret key is c and the server's long-term secret key is s. The cache's long-term public key is then Curve25519(c), and the server's long-term public key is Curve25519(s). The cache and the server both compute a shared secret Curve25519(cs), and then use fast secret-key cryptographic mechanisms to encrypt and authenticate data.

DNSCurve does not use signatures broadcast from one public key. Signatures might seem to be an adequate substitute for two-key protection when confidentiality is not required, and they would allow an important speedup: the server, after computing a signature once, can reuse the signature for any number of clients. However, DNSCurve allows two speedups that turn out to be even more important:

- The server, after computing the secret shared with a particular cache, can reuse the secret for any number of packets exchanged with that cache.
- The cache, after computing the secret shared with a particular server, can reuse the secret for any number of packets exchanged with that server.

In the paper describing Curve25519[33], the following explanation is given (cited):

Here is the high-level view of Curve25519: Each Curve25519 user has a 32-byte secret key and a 32-byte public key. Each set of two Curve25519 users has a 32-byte shared secret used to authenticate and encrypt messages between the two users.

Medium-level view: Figure 8 shows the data flow from secret keys through public keys to a shared secret.

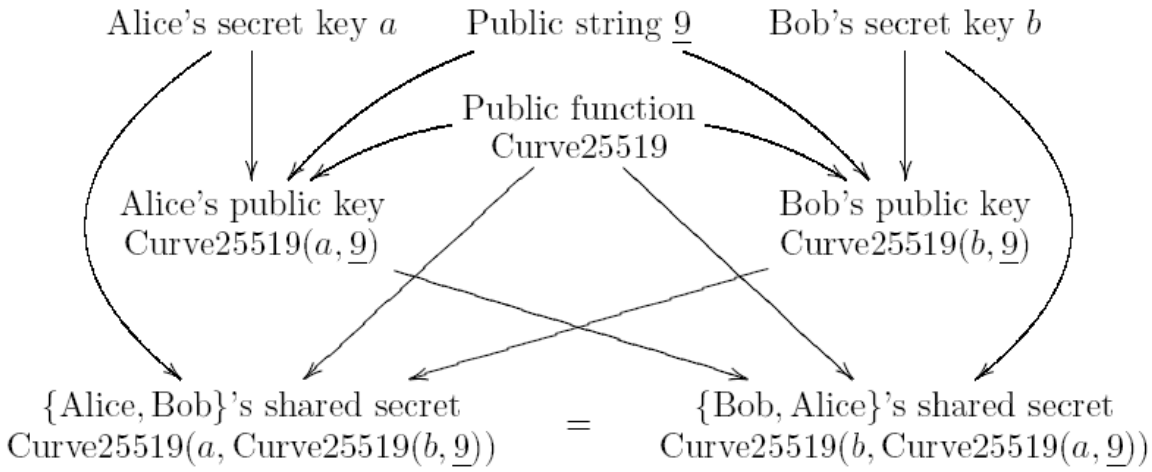


Figure 8. Overview of the dataflow within Curve 25519. Courtesy of D. J. Bernstein.

A hash of the shared secret Curve25519(a; Curve25519(b; 9)) is used as the key for a secret-key authentication system (to authenticate messages), or as the key for a secret-key authenticated-encryption system (to simultaneously encrypt and authenticate messages).

7.2 DNSSEC Cryptographic Algorithms

The DNS security extensions are designed to be independent of the cryptographic algorithms used. The DNSKEY, RRSIG, NSEC3 and DS Resource Records have an algorithm field to identify the cryptographic algorithm used. RFC 4034 specifies that a DNSSEC aware resolver or name server must implement all mandatory algorithms, however at the time of writing there is only one mandatory algorithm (RSA/SHA1). Table 3 shows the available DNSSEC cryptographic algorithm types.

Number	Algorithm	Mnemonic	Signing	Status	Reference
0	Reserved				RFC 4398
1	RSA/MD5	RSAMD5	N	NOT RECOMMENDED	RFC 4034 RFC 2537
2	Diffie-Hellman	DH	N		RFC 2539
3	DSA/SHA1	DSA	Y	OPTIONAL	RFC 3755 RFC 2536
4	Elliptic Curve	ECC			
5	RSA/SHA1	RSASHA1	Y	MANDATORY	RFC 3755 RFC 3110
6	DSA-NSEC3-SHA1	DSA-NSEC3-SHA1	Y		RFC 5155
7	RSASHA1-NSEC3-SHA1	RSASHA1-NSEC3-SHA1	Y		RFC 5155
8 – 251	Unassigned				
252	Reserved for Indirect Keys	INDIRECT	N		RFC 4034
253	Private algorithms – domain name	PRIVATEDNS	Y	OPTIONAL	RFC 3755 RFC 2535
254	Private algorithms – OID	PRIVATEOID	Y	OPTIONAL	RFC 3755 RFC 2535
255	Reserved				RFC 4034
Table based on: RFC 4034, RFC 5155, IANA Domain Name System Security (DNSSEC) NextSECure3 (NSEC3) Parameters IANA Domain Name System Security (DNSSEC) Algorithm Numbers					

Table 3. The available DNSSEC cryptographic algorithm types.

Cryptographic algorithms 6 and 7 are used with NSEC3 Resource Records as described in RFC 5155 and are actually aliases for DSA and RSA.

As shown in the table above there is already a number reserved for the use of Elliptic Curve Cryptography (ECC). Within the Internet Engineering Task Force (IETF) there are discussions about ECC (draft-ietf-dnsext-ecc-key[34]). This Internet-Draft expired August 2007. However it is interesting when DNSSEC should be capable of using ECC that will reduce much of the differences between DNSCurve and DNSSEC. The following sentence is cited from the draft:

> "*Elliptic curve signatures use smaller moduli or field sizes than RSA and DSA. Creation of a curve is slow, but not done very often. Key generation is faster than RSA or DSA.*"

This could speed up the key generation instead of using RSA; it would be interesting to see ECC within DNSSEC and which curve they would use.

8. Threats towards implementation

This chapter summarizes the threats that DNSCurve and DNSSEC are facing towards implementations. First the separate threats will be discussed followed by a more general section and at the end some results about the adoption of DNSSEC.

- **Backward compatible**
  Improvements may not influence other technologies that rely on the existing DNS standards. This can prevent fundamental changes, but guarantee that older hardware and software keeps working.

- **Protocol stability**
  The protocol should remain untouched if modifications to the DNS protocol require re-implementation of the DNS server and client code.

- **No radical changes**
  It turned out that large-scale replacements or modifications to the existing DNS infrastructure can be showstoppers for the implementation, because they come with a certain uncertainty.

8.1 Threats that DNSCurve faces

- **Lack of a formal specification.** Now there is only a website with some facts and highlights.
- **Lack of implementation ready software.** The available software is outdated (2008-09-18) and contains errors that withheld organizations from implementing it. There seems to be some activity on the website (2009-06-26).
- **Lack of installation manuals**. Making it even harder to install the software.
- **Not clearly specified which problems / shortcomings of DNS it solves.** Thereby making it difficult to evaluate how well DNSCurve succeeded.
- **Uses a specific elliptic curve.** Curve25519 that is modified by D. J. Bernstein. Though he has a reputation of writing secure software the modifications should be verified by other cryptographic experts.
- **No cryptographic algorithm rollover mechanism.** If the algorithm fails or become insecure. DNSSEC has OPTIONAL defined algorithms that might be implemented.
- **Encryption at the transport layer**. That by itself does not ensure integrity.
- **NS record hack.** By some people considered as type overloading.
- **Labels exceed 63-bit limit.** When encoding the public key in the NS record, the combination with the existing label may exceed the 63-bit limit.

8.2 Threats that DNSSEC faces

- **Complexity.** This remains a problem but is getting better as more organizations implement it and offer support.
- **Introducing new problems.** Possible Denial of Service (DoS) due to amplifying situation in traffic, zone walking and Next Secure (NSEC) records (partly solved with NSEC3 RFC 5155).
- **Impact on cache servers,** Cache servers must also validate the responses; this might have a serious impact on the performance.
- **Problems with the so called "middle boxes"** (firewall / NAT). The traffic is not always handled well through these "middle boxes".
- **Protocol overhead.** There will be severe overhead compared with the DNS as we know it today.
- **Annual and monthly key rollover.** This need to be well tested before performing a key rollover. Waiting is not an option as it would result in expired RRsets, which can led to inaccessible zones.
- **Lack of experience with NSEC3 Resource Records.** Some experience available from the signing of the .org TLD.

- **Dynamic updates are not possible anymore.** However the NIST[35] says: "DNSSEC can be used with dynamic update, as long as the signing key is on the server". This does not seem to be conform common security practices; private keys should be kept offline. Although one could use multiple signing keys. It is still giving up security for functionality.
- **The zone file grows approximately 7 times.** Depending on the DNS software that is used there might be other requirements, like additional memory.
- **Increased bandwidth.** Larger responses must be transmitted.
- **Time required for cryptographic key-pair generation** can be problematic for organizations with many zones.
- **Lack of DNSSEC validating resolvers in Operating Systems (OS).** This prevents an end-to-end DNSSEC validation.
- **Small Office Home Office (SOHO) routers do not always handle DNSSEC traffic well.** In this report[36] some SOHO routers have been tested. This might be a chicken-egg problem because there are not much DNSSEC validating resolvers in OSs and customers often use the DNS server of their Internet Service Provider (ISP).

8.3 General threats

- **Why change our DNS? It works now.** Fear to change a working DNS environment and requiring more administration and knowledge.
- **Nowadays the DNS configuration is very static**. Zone files do not change very often. When implementing DNSCurve or DNSSEC more administration is necessary.
- **Partial solutions possible**. For example: Interim Trusted Anchor Repositories (ITARs) are generating islands-of-trust. Solutions that do not protect the root zone have limited result.
- **Interim security improvements.** Solutions primarily used to prevent cache poisoning but not securing the root, for example: DNS-0x20 encoding, UDP port randomization, and DNS resolvers with birthday protection.
- **Presumable more costs for customers.** Costs for new hardware, implementation, more administration and external expertise.
- **Not all customers might want to deploy the techniques.** Depending on the organization, it might not be a potential interesting attack target. However it might also divide rich and "poor" organizations.
- **Embedded devices.** Sometimes they cannot be upgraded or are not powerful enough to perform cryptographic operations.
- **Routers that do not support EDNS.** This limits UDP messages to 512 bytes and would cause truncation.

8.4 Politics

Although politics are not part of the research, they keep coming back in discussions and standardization Working Groups (WG). It can be compared with something like this: "Either you are with us (DNSSEC) or you are against us", sometimes being an author (in this case D. J. Bernstein) of a piece of software seems to be enough to be ignored. It is in the interest of the Internet that this people should attack each other based on their proposals, not on who they are. Below are some quotes that I found on the internet:

> "Bind Cartel" [37]
> "I still need to punch him in the face for qmail" [38]

Expiration / Renewal of JPA
There is an upcoming opportunity to change the governance of the Internet, due to the expiration of the Joint Project Agreement[39] (JPA) in September 2009. This agreement creates the relationship between the United States Government (Department of Commerce and the National Telecommunications and Information Administration (NTIA)), Internet Assigned Numbers Authority (IANA lead by the ICANN) and VeriSign. The existing Internet governance relies on this agreement and the 13 DNS root servers results of the trust in this authority.

From different sides the influence of the United States Government is criticized, even from within the European Union[40]. This has nothing to do with the technical aspects, but more with the transparency of governance. In France there is an initiative called Net4D[41] (Networks for Development) which is complementary with the existing DNS. It might be interesting what happens when discussions about the JPA turns out to be useless. Below there is a short description from the website of Net4D:

> "*New classes of networks to bind people and machines*
> *The Next Generation of Domain Names Services*
> *A new opportunity for scientific, cultural, linguistic and economic development*"

Root signing
Recent announcements[42] from the ICANN shows that they will work with the U.S. Department of Commerce's, National Telecommunications and Information Administration (NTIA), the National Institute of Standards and Technology (NIST) and VeriSign on the goal of an operationally Signed Root Zone as soon as feasible in 2009. The discussion here is: who will hold the keys of the root zone. Again the United States Government comes in place here, because of the power associated with this keys it is not likely that the U.S. Government is willing to give up its position soon.

The details of the process are still being worked on but discussions between the Department of Commerce, VeriSign and ICANN have identified that VeriSign will manage and have operational responsibility for the Zone Signing Key (ZSK) in the interim arrangement, and that ICANN will manage the Key Signing Key (KSK) process.

8.5 DNSSEC deployment

SecSpider
There are several websites that actively monitor the DNSSEC deployment, an interesting one is the SecSpider[43] project. SecSpider is a globally based polling system that crawls through a list of secure zones every day. The pollers are distributed around the globe to preserve that the observed data is consistent from various locations; this also prevents the polling from local connection problems. Below is an example of the collected information:

Deployment status as of: Thu Jul 2 00:56:41 2009 UTC

Monitoring Summary:
18092 Zones

16406 Zones have NS sets that match their parents' delegation set
11983 DNSSEC enabled zones

4572 Zones use both KSKs and ZSKs
3476 Production DNSSEC-enabled zones

World Wide DNSSEC Deployment map
Another example is the World Wide DNSSEC Deployment map[44] (based on Google maps). The next page shows a screenshot (figure 9) of the deployment map. The map itself shows the following DNSSEC related information:

- TLD Production
- Reverse Production
- ccTLD Testbeds
- gTLD Testbeds
- DLV Registry
- Unofficial Projects
- Discontinued

Figure 9. The World Wide DNSSEC Deployment map. Courtesy of P. Wouters.

DNSSEC Logo

To increase the awareness and visibility of DNSSEC there is even a DNSSEC Logo[45] that a DNS administrator can obtain. Depending on the degree of implementation there is a Bronze, Silver or Gold logo.

ENISA

The European Network and Information Security Agency (ENISA) interviewed[46] several European Service Providers about their experience with DNSSEC. Below are the results showing the deployment status and challenges (including the questions). Figure 10 shows the deployment status under European Service Providers.

"*Have you implemented DNSSEC or do you plan to implement it in the next 2-3 years?*"

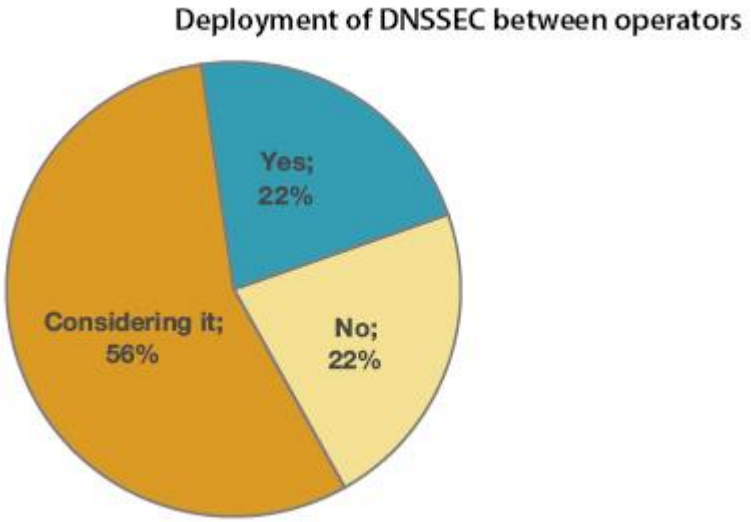


Figure 10. Deployment of DNSSEC under European Service Provider. Courtesy of ENISA.

Almost a forth (22%) do not plan to deploy DNSSEC within the next 3 years. Interestingly are the reasons they name: lack of customer demand for the service. Other reasons are the cost (initial / on-going) and one other mentioned the lack of requirements from national regulators.

Also 22% have already deployed DNSSEC in their DNS services, but the majority (56%) is considering deployment within the next three years. The main reason they name is the improvement in the resilience of the DNS.

Figure 11 shows a graph of the challenges a DNSSEC implementation faces, based on the question below:

> "*What barriers, if any, do you/did you see for DNSSEC deployment? (e.g., zone walking NSEC/NSEC3, key management complexity, cost, lack of signed root zone)*"

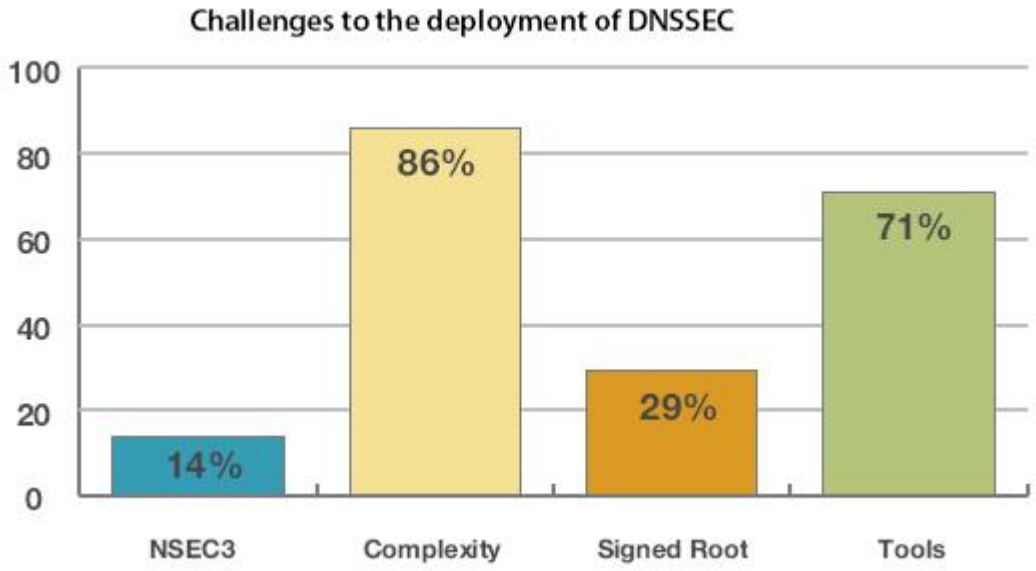


Figure 11. The biggest DNSSEC deployment challenges. Courtesy of ENISA.

Complexity still remains the biggest challenge (86%) when implementing DNSSEC. This also comes from the lack of tools (71%) for automating operations. 29% Mentioned the lack of a signed root in combination with the lack of common policies for Trust Anchor distribution and update. The delay of the introduction of NSEC3 Resource Records is seen as challenge by 14%.

9. Tools

Tools are necessary to automate manual operations and reduce complexity. Especially when manual operations need to be conducted on a frequently basis, the risk of human errors and misconfiguration increases. Errors and misconfiguration can result in serious service outages, like unreachable zones.
The size of the DNS infrastructure and the amount of changes are important factors when choosing a tool. However it should be noticed that there is no a one-size-fits-all solution and that zone administrators often have different requirements.

9.1 DNSCurve

The DNSCurve website does not provide any information about tools that can be used for installation and monitoring. Presumable the normal tools for managing the DNS infrastructure can be used, however this should be tested as the software comes available. Due to the different approach of DNSCurve it does not need to do key-rollover, (re)-signing etc. Operational tasks should remain the same and could at least theoretically, use the same tools as before.

9.2 DNSSEC

Tools[47] for DNSSEC are more commonly available. Some tools are even usable without having DNSSEC implemented. There are some graphical tools; however the command line support is usually better. On the other hand graphical tools might be useful for new users without much command line experience and they can be useful to keep a good graphical overview.

The categorization as shown below is based on the tools available on the DNSSEC-Deployment website[48]. There are simply too many tools to describe each one of them, besides the list may be updated frequently and tools get updated.

More tools can be found in the DNSSEC-Tools package[49], this package contains many tools that can be used with DNS and DNSSEC administration. The paper “Using DNSSEC-Tools to Deploy DNSSEC” [50] gives an overview of using existing tools and utilities from the DNSSEC-Tools suite to build environments that support DNSSEC along the complete path from the authoritative name server where domain data resides to the end-application that uses DNS data.

9.2.1 DNSSEC tool categorization

1. **Zone administration tools**

Name servers
: Different DNS name server distributions and types: authoritative, validating, recursive, caching etc.

Key generation and zone signing
: Key generating and zone signing tools that are part of different DNS distributions.

Key rollover
: Managing different phases of ZSK and KSK rollover.

Hardware related
: Different hardware solutions and extensions for DNSSEC.

Zone troubleshooting
: Tools for testing zone files (contents), visual mapping, verifying signatures etc.

2. Secure delegation registration

Creation of DS information
Tools to generate Designated Signer (DS) from DNSKEY records.

Updating DS from child to parent
Tools for updating contents of a registry and moving keys from sub-zones to parent-zones.

3. Tools for validating systems

Tools and resources for fetching DNSKEY information
Tools for constructing and populating Trust Anchor Repositories (TAR). Tools for receiving, fetching and comparing and DNSKEYs from a domain.

Tools for automated trust anchor rollover
Tools that implement RFC 5155 for automated rollover of trust anchors in validating resolvers.

Troubleshooting
Tools that can be used for troubleshooting, visualization of DNS packet flows etc.

DNSSEC capable applications
Patches that enable DNSSEC validation in different application, for example: Firefox, Thunderbird, SSH, wget etc.

4. Developer resources

Validation libraries for applications
Different libraries that provides validation capabilities.

Perl SDKs
Perl Software Development Kits (SDKs) that provide different extensions.

Validator API
Different validator Application Programming Interfaces (APIs).

Testing resources
Tools for generating test data that can be used against DNSSEC aware software, replay tools etc.

5. Deployment aids

Operator guidance documentation
Deployment guides, operational practices guides, tutorials etc.

10. Interim solutions

Interim solutions can fill the gap between the existing DNS infrastructure and the implementation of DNSCurve or DNSSEC. There are several techniques and proposals that can be used to increase the security level, some of them focus on the name server where others focus on the resolver. Some of these techniques are already standardized by the IETF while others are still a draft. The use of interim solutions is often under discussion because it can split the implementation of DNSSEC, for example when the islands-of-trust remain to exist. Some of the more interesting solutions are discussed.

10.1 ITAR

IANA provides an Interim Trust Anchor Repository[51] (ITAR) to share the key material required to perform DNSSEC verification of signed Top Level Domains. The optimal scenario where the entire DNS tree is signed and the only Trust Anchors that the validating resolver needs to know are the Secure Entry Point (SEP) keys for the root zone. The root zone will not be signed before the end of 2009, resulting in a fragmentation of islands-of-trust. Where the sub-tree contains signed zones but the parent remains unsigned. With as result that DNSSEC validating resolvers are not able to verify the authentication by using the chain-of-trust, unless each island-of-trust has a configured Trust Anchor. This would require that every DNSSEC validating resolver keep track of all the SEP keys for each island-of-trust, this is highly unlikely and very impractical.

A Trust Anchor Repository[52] can be seen as a DNS Resource Record store that contains SEP keys for multiple zones. It offers the validating resolver the possibility to fetch Trust Anchor information for a number of zones without having to manage all the information locally. This is a temporary service that will stop when the root zone is signed, from there on the keying material will be placed in the root zone.

10.2 DLV

Domain Lookaside Validation (DLV) provides an alternative for the lack of signed TLD zones. DLV enables one or more alternative chain-of-trust and functionally identical domain authentication without the need for any of the TLDs to have signed zone files. DLV is standardized in RFC 5074[53].

10.3 DNS-0x20 encoding

The limited size of the DNS transaction ID, 16-bit (65536 possibilities) made it an easy target for forgery, resulting in many cache poisoning vulnerabilities. Even when the transaction ID's are unpredictable, random and birthday attacks are still theoretically feasible. DNS-0x20[54] [55] encoding describes a simple and practical technique to make DNS queries more resistant against poisoning attacks: mix the upper and lower case spelling of the domain name in the query. See the example below:

Normal: www.example.com
DNS-0x20: WwW.eXaMplE.cOm

Almost all DNS name servers preserve the mixed case encoding of the query in the response messages, for example when we look at the os3 name server:

```
dig @ns1.os3.nl WwW.oS3.NL

; <<>> DiG 9.4.2-P2 <<>> @ns1.os3.nl WwW.oS3.NL
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6089
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;WwW.oS3.NL.                    IN      A

;; ANSWER SECTION:
WwW.oS3.NL.             86400   IN      CNAME           info4u.oS3.NL.
info4u.oS3.NL.          86400   IN      A               145.100.96.70
<SNIP>
```

Attackers that want to poison the DNS cache must now guess the mixed-case encoding of the query, in addition to all other fields required in a DNS poisoning attack. This increases the entropy, but depends on the length of the domain name. The drawbacks here are short domain names, for example www.nu.nl and numerical domain names www.112.nl. The options here are limited because only “a..z” and “A..Z” can be encoded and there are just a few possibilities.

10.4 ENDS-PING

This draft[56] from A. Hubert describes an EDNS-PING which in effect allows a far longer DNS transaction ID, making it much harder for an external attacker to inject bogus responses. With ENDS-PING the remote name server is asked to copy a certain string from the query to the response. This string can be used to verify the proper transmission of DNS queries and responses of different sizes.

The drawback of this solution is that it does not offer protection against in-line attackers with the ability to not only inject responses, but to modify existing ones or intercept questions and inject tailored responses. But for almost all solutions (without encryption) in-line attackers can modify the traffic or simply drop it, resulting in unavailable DNS services.

10.5 Resolver side mitigation

This draft[57] from W. Wijngaards describes a set of mitigations that stop the known variations of the Kaminsky attack against the DNS system, for which only resolver side deployment is necessary.
Notice that this draft focuses on Unbound.

10.6 TSIG

Transaction Signature (TSIG) is specified in RFC 2845[58] and is a protocol that can be used for transaction level security using shared secrets and one way hashing. Within DNS it can be used to authenticate dynamic updates verifying they come from an approved client or it can be used to authenticate responses, verifying they come from an approved recursive name server.

The protocol does not describe mechanisms for distributing the shared secrets. According to the RFC; they expect the network administrator to statically configure name servers and clients using some out of band mechanism such as sneaker-net until a secure automated mechanism for key distribution is available.

Well here are some problems: shared secrets and the distribution. All the name servers and clients must share the same secret making it much easier to expose the secret and with that the whole security, requiring everyone to update the shared secret. The other problem is the shared key

distribution; this might work in a small company but does not scale outside the company. To preserve some level of security out of band mechanisms are necessary but they can be intercepted as well resulting in the expose of the whole security.

In essence one could say that TSIG does the same as DNSCurve: providing transport security. This is true but DNSCurve is implemented on a completely different manner.

The existing TSIG RFC is updated by Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS TSIG) RFC 3645[59].

11. Conclusions

The main research question was the following:

> " *What consequences do the differences in design of DNSCurve and DNSSEC have on the implementations?* "

The most important difference between DNSCurve and DNSSEC is the way they protect the Domain Name System. DNSCurve offers authentication and encryption to the link-layer, whereas DNSSEC offers message authentication and integrity verification through cryptographic signatures. This difference is essential to the design of both techniques.

Both DNSCurve and DNSSEC require changes in the existing DNS infrastructure, although the DNSCurve stand-alone forwarder should work without changes to the existing DNS server. DNSCurve claims to be simple and easy installable whereas DNSSEC is being known for its complexity.

Due to the signature based approach, all records within a zone need to signed, including the "non-existing" records, DNSCurve does not sign records. Both techniques are somewhat similar to a Public Key Infrastructure (PKI), where DNSCurve uses "overloading" of the existing NS records and DNSSEC introduces a new DNSKEY record to store the public key. DNSCurve uses elliptic curve cryptography and DNSSEC uses the well known RSA algorithm. Based on to the available literature DNSSEC seems to consume the most resources, due to: increased zone file, bandwidth increase, computing power to generate keys and perform signing of the records.

Partial deployment is possible with DNSCurve and DNSSEC, but the latter requires the use of an Interim Trust Anchor Repository (ITAR) as long as the root zone is not signed. These ITARs will stop as soon as the root zone is signed.

DNSCurve is relatively new and there are no known implementations instead of the more mature DNSSEC. Based on the information about signing the root zone, the signing of the .org TLD and several initiatives with ccTLD, the demand that US Government organizations need to install DNSSEC, etc. it is likely that they will continue with DNSSEC instead of DNSCurve.

DNSCurve seems to be very promising but first have to prove itself.

12. Future work

The scope of this research is limited to a theoretical investigation. Simply because there was no actual production ready DNSCurve software available at the time of this writing. During the last week of the research period the DNSCurve website was renewed, updated and more important: it shows that the software is being developed and tested. When the DNSCurve software is released there is some interesting research possible. Below are some interesting aspects of DNSCurve and DNSSEC that might be investigated.

- **Formal specification**, due to the lack of a formal specification it is hard to get a clear understanding of DNSCurve. There is a website, but that is more a collection of facts and some highlights. It would be interesting to see some kind of formal specification.

- **DNSCurve code analysis**, how is the encryption algorithm implemented etc. There is some old code available, but it is not clear if and how that differs from the official released software.

- **DNSCurve versus DNSSEC tests**, when DNSCurve is official released performance, scalability, etc tests can be conducted. This can be done using different data sets, however it should be noticed that they protect the DNS on a different manner.

- **Cryptographic analysis of Curve25519**, let cryptographic experts look at the modified curve, as with all cryptographic algorithms.

- **DNSCurve through firewalls**, how will firewalls handle DNS traffic on port 53 when it is encrypted, especially firewalls with packet inspection. Normal DNS and DNSSEC traffic can exceed the 512 bytes limit, but it is not encrypted.

- **Impact on embedded devices**, can they be updated to DNSCurve and DNSSEC and more important, do they have enough computing power to do cryptographic calculations.

- **DNSSEC through SOHO routers**, many customers use the DNS server of their ISP, but what happens when they want to do the validation themselves. Do SOHO routers support DNSSEC traffic well? There is some research on this topic but when implementing DNSSEC on a large scale it should get more attention. However this depends on the availability of DNSSEC validating resolvers in the operating systems.

- **DNSTrust**, the DNSTrust[60] websites shows the trust connections for TLDs. More implicit trust dependencies for a TLD makes a TLD more vulnerable to DNS cache poisoning. The website gives an example attack where one of the implicitly trusted name servers takes over control of the entire .fr domain. The website also states that this can be prevented by implementing DNSSEC. Interesting to investigate the trust dependencies for the .nl TLD.

- **DNSSEC and IPv6**, through the use of IPv6 DNS messages might also increase. This might have consequences for DNSSEC.

- **DNSSEC validating resolvers in OSs**, when creating end-to-end validation, operating systems must include DNSSEC validating resolvers. Windows 7 comes with a DNSSEC validating resolver; it is interesting to look at the DNSSEC handling and what to do when the validation fails, stop sign, SSL like methods etc.

- **Key revocation**, when the private key gets compromised immediate key revocation, rollover and signing is necessary. DNS caching servers that are managed by the organization can be flushed but other servers have to wait till the TTL expires, so forged data can remain in caches for some time. There is a paper[61] discussing this problem, however more research on this topic is necessary.

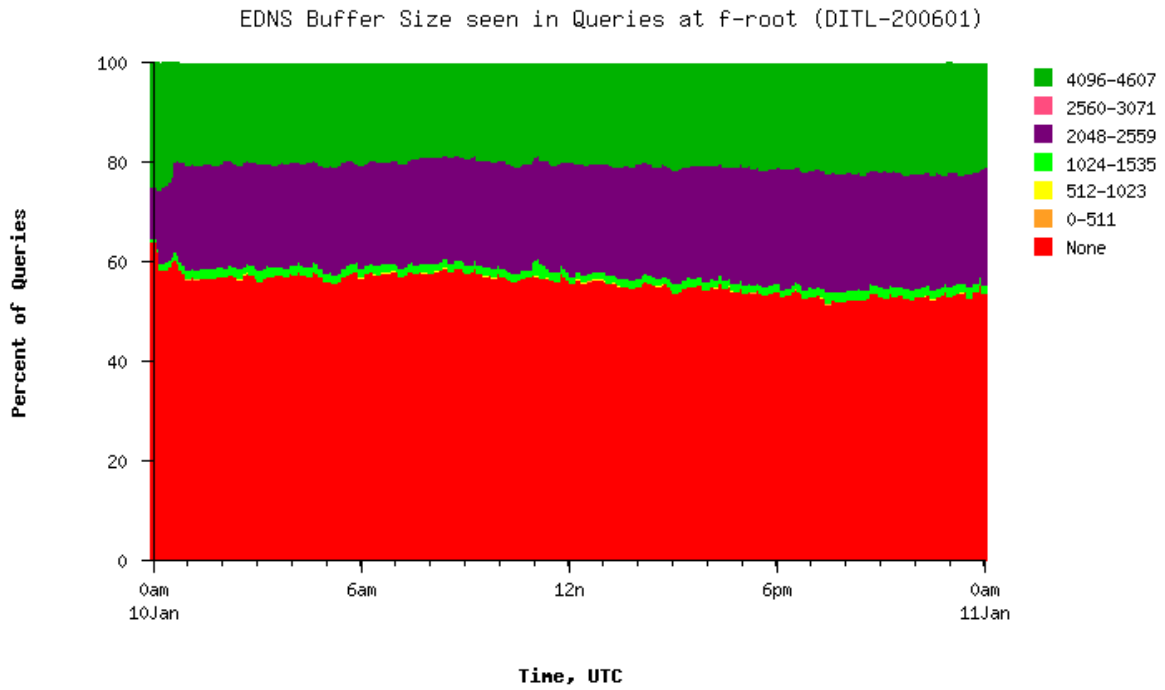
Acronyms

AD	Authenticated Data
ARPA	Advanced Research Projects Agency
CACE	Computer Aided Cryptography Engineering
ccTLD	County Code Top Level Domain
CD	Checking Disabled
DLV	DNSSEC Lookaside Validation
DNS	Domain Name System
DNSKEY	Domain Name System Public Key
DNSSEC	Domain Name System Security Extensions
DS	Designated Singer
ECC	Elliptic Curve Cryptography
EDNS0	Extension Mechanisms for DNS
FP7	Seventh Framework Programme
gTLD	Generic Top Level Domain
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
IETF	Internet Engineering Task Force
ISP	Internet Service Provider
ITAR	Interim Trust Anchor Repository
KSK	Key Signing Key
NaCl	Networking and Cryptography library also known as (salt)
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NSEC	Next Secure
NSEC3	Next Secure3
NTIA	National Telecommunications and Information Administration
PKI	Public Key Infrastructure
RFC	Request For Comment
RIPE-NCC	Réseaux IP Européens Network Coordination Centre
RR	Resource Record
RRset	set of Resource Records
RRSIG	Resource Record Signature
SEP	Secure Entry Point
TCP	Transmission Control Protocol
TLD	Top Level Domain
TSIG	Transaction Signature
TTL	Time To Live
UDP	User Datagram Protocol
ZSK	Zone Signing Key

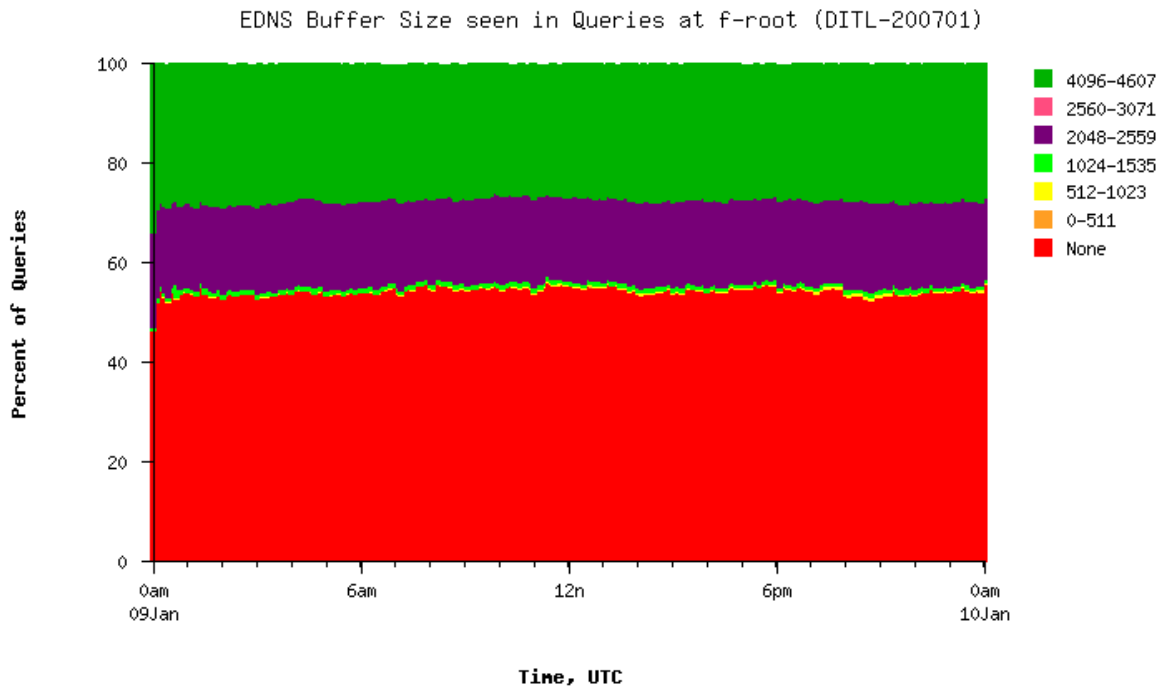
Appendix 1 EDNS Buffer sizes at f-root

The figures below show the increasing EDNS0 support and the different UDP message sizes in use from 2006 till 2008. Interesting to see is the amount of message sizes (2048) that is decreasing through the years. Notice these figures are from the f-root servers. All the figures are courtesy of the DNS Operations, Analysis, and Research Center (DNS-OARC).

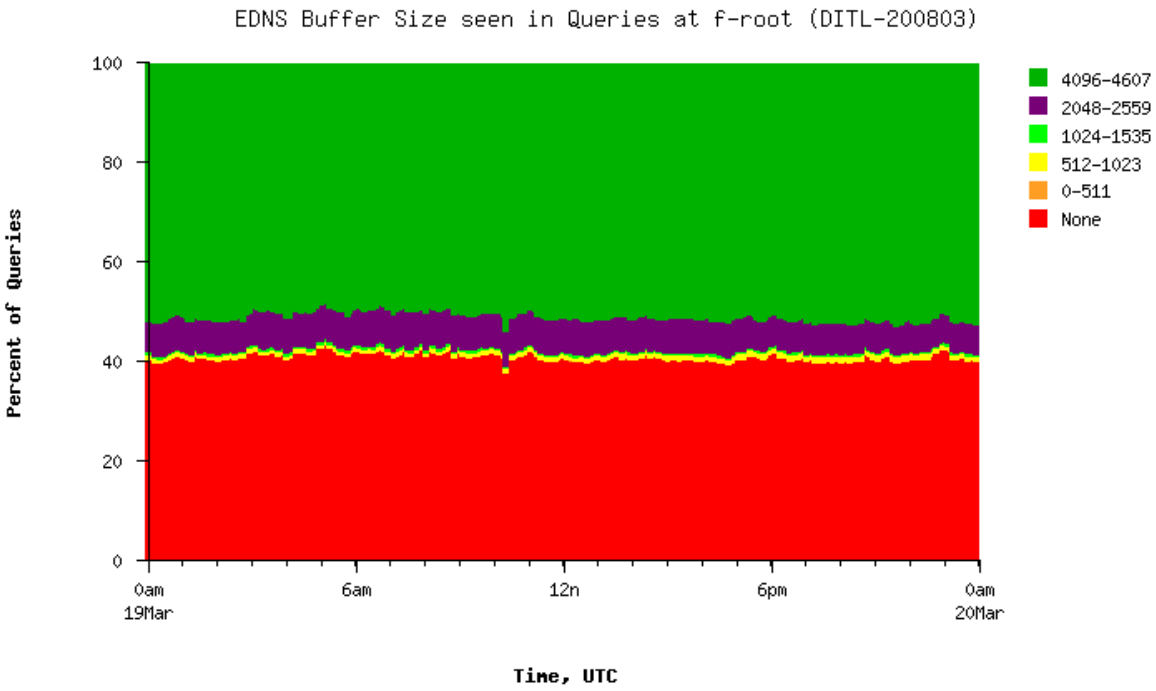
January 2006



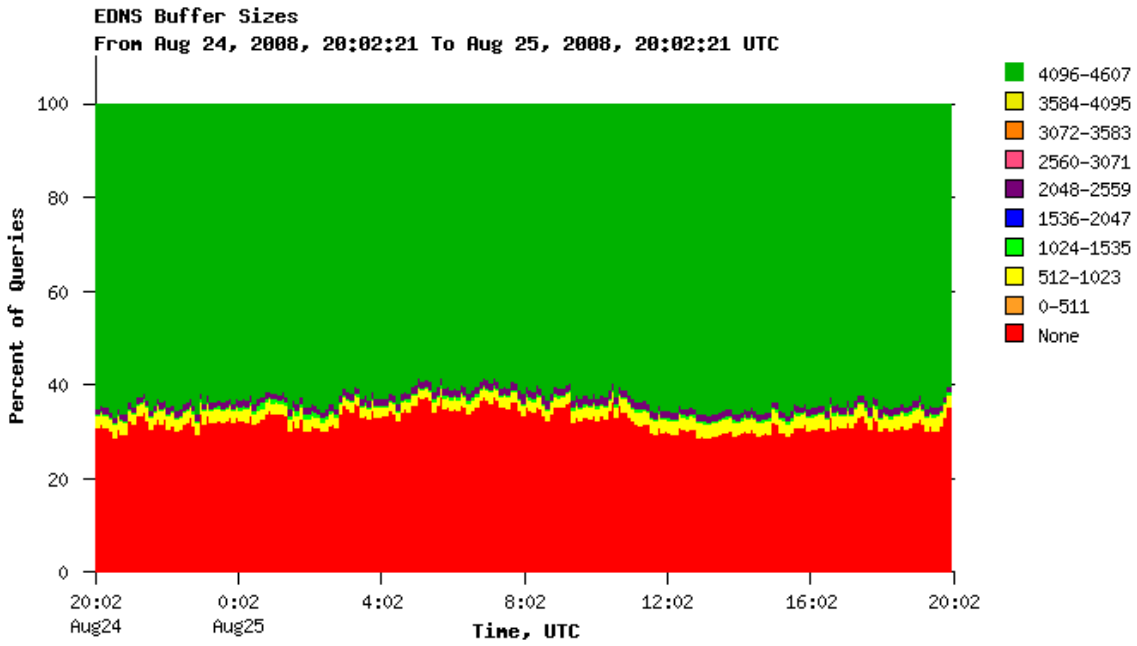
January 2007



March 2008

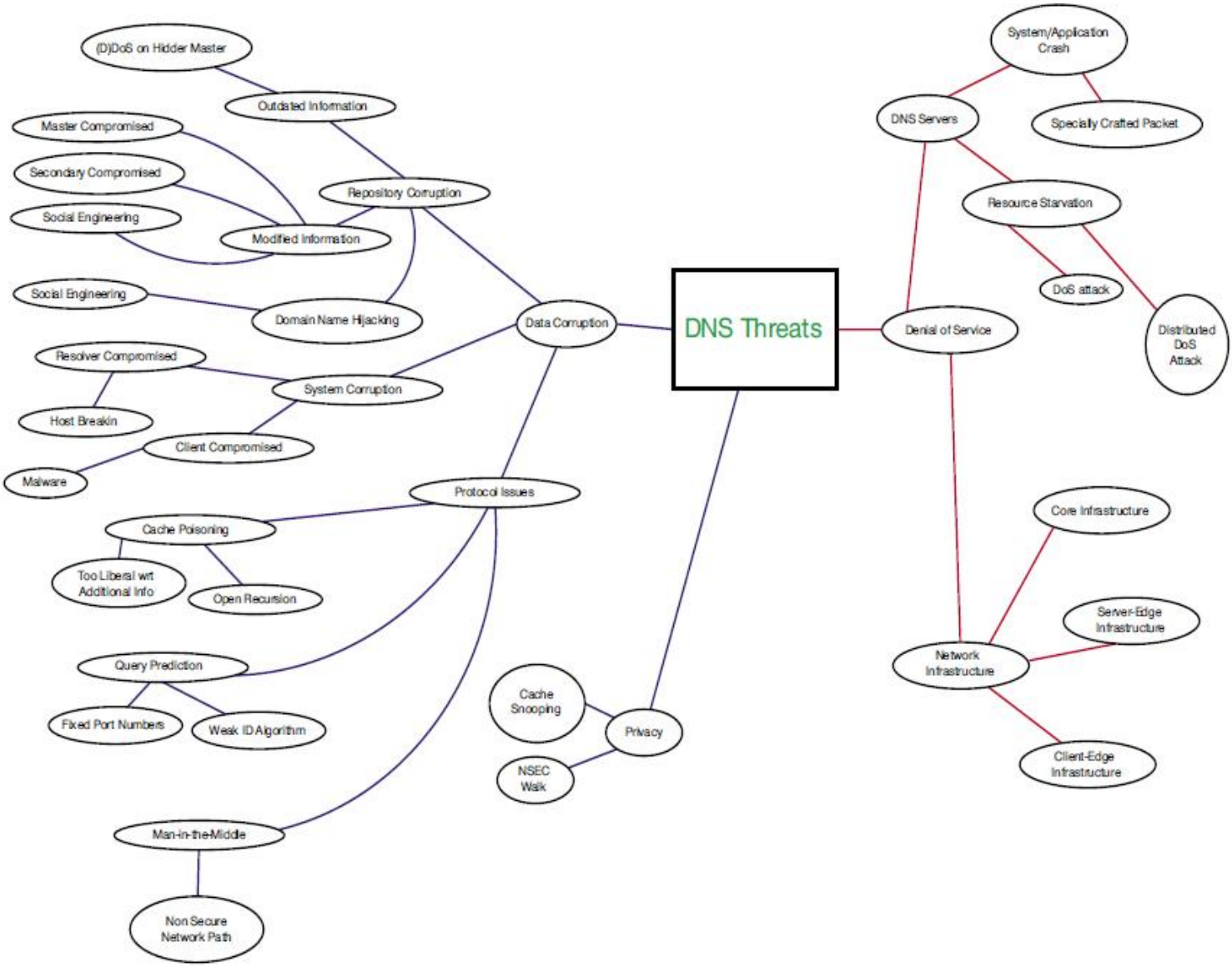


August 2008



Appendix 2 Main threats against the DNS

The figure below shows the main threats against the DNS. The threats on the right side of the figure cannot be addressed by DNSSEC. Courtesy of ENISA.



Literature

[1] P. Mockapetris, “Domain names – Concepts and facilities”, RFC 1034, November 1987.
URL http://tools.ietf.org/html/rfc1034

[2] P. Mockapetris, “Domain names – Implementation and specification”, RFC 1035, November 1987.
URL http://tools.ietf.org/html/rfc1035

[3] M. Lottor, “Domain administrators operations guide”, RFC 1033, November 1987.
URL http://tools.ietf.org/html/rfc1033

[4] What is a fully qualified domain name (FQDN)?, Indiana University, University Information Technology Services, Knowlede base, May 2009.
URL http://kb.iu.edu/data/aiuv.html

[5] A. S. Tanenbaum, M. van Steen, “Distributed Systems: Principles and Paradigms, Second Edition”, Prentice Hall, Upper Saddle River, NJ, 2007.

[6] S. M. Bellovin, “Using the Domain Name System for System Break-ins” in: Proceedings of the fifth USENIX UNIX Security Symposium, Salt Lake City, Utah, June 2005.
URL http://portal.acm.org/citation.cfm?id=1267591.1267609

[7] D. Atkins, R. Austein, “Threat Analysis of the Domain Name System (DNS)”, RFC 3833, August 2004.
URL http://tools.ietf.org/html/rfc3833

[8] D. Eastlake and C. Kaufman, “Domain Name System Security Extensions”, RFC 2065, January 1997.
URL http://tools.ietf.org/html/rfc2065

[9] D. Eastlake, “Domain Name System Security Extensions”, RFC 2535, March 1999.
URL http://tools.ietf.org/html/rfc2535

[10] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “DNS security introduction and requirements”, RFC 4033, March 2005.
URL http://tools.ietf.org/html/rfc4033

[11] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “Resource records for the DNS security extensions”, RFC 4034, March 2005.
URL http://tools.ietf.org/html/rfc4034

[12] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “Protocol modifications for the DNS security extensions”, RFC 4035, March 2005.
URL http://tools.ietf.org/html/rfc4035

[13] B. Laurie, G. Sisson, R. Arends and D. Blacka, “DNS Security (DNSSEC) Hashed Authenticated Denial of Existence”, RFC 5155, February 2008.
URL http://tools.ietf.org/html/rfc5155

[14] An Astonishing Collaboration, Personal blog of D. Kaminsky, July 2008.
URL http://www.doxpara.com/?p=1162

[15] DNSCurve: Usable security for DNS, (Notice: website frequently updated since 2009-06-22).
URL http://dnscurve.org/

[16] NaCl: Networking and Cryptography library, March 2009.
URL http://nacl.cace-project.eu/

[17] CACE: Computer Aided Cryptography Engineering.
URL http://cace-project.eu/

[18] FP7: Seventh Framework Programme, July 2009.
URL http://cordis.europa.eu/fp7/home_en.html

[19] O. M. Kolkman, “DNSSEC Raising the Barriers against DNS exploits”, NLnet Labs, November 2005.
URL http://www.nlnetlabs.nl/downloads/RaisingTheBarrier.pdf

[20] P. Brand, R. van Rein, R. van Rijswijk and D. Yoshikawa Hardening the Internet, "The impact and importance of DNSSEC", SURFnet, January 2009.
URL http://www.surfnet.nl/Documents/DNSSSEC-web.pdf

[21] Tools for DNS curve implementation, Github Social Coding, September 2008.
URL http://github.com/agl/dnscurve/tree/master

[22] M. Timmers, "DNSCurve analysis", Universiteit van Amsterdam, February 2009.
URL http://staff.science.uva.nl/~delaat/sne-2008-2009/p32/report.pdf

[23] ICANN, "Survey of DNSSEC Capable DNS Implementations", SAC 030, July 2008.
URL http://www.icann.org/en/committees/security/sac030.htm

[24] NIST, "DNSSEC and its Impact on DNS Performance".
URL http://www-x.antd.nist.gov/dnssec/dnssec-perform.html

[25] O. M. Kolkman, "Measuring the resource requirements of DNSSEC", RIPE NCC / NLnet Labs, September 2005.
URL http://www.ripe.net/ripe/docs/ripe-352.html

[26] A. Guillard, "DNSSEC Operational Impact and Performance", British Telecommunications plc, August 2006.
URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4124082

[27] TechNet, "Using DNS Security Extensions (DNSSEC)", Microsoft, January 2005.
URL http://technet.microsoft.com/en-us/library/cc728328(WS.10).aspx

[28] TechNet, "An Introduction to Security in Windows 7", Microsoft, May 2009.
URL http://technet.microsoft.com/en-us/magazine/2009.05.win7.aspx

[29] P. Vixie, "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.
URL http://tools.ietf.org/html/rfc2671

[30] D. Conrad "Indicating Resolver Support of DNSSEC", RFC 3225, December 2001.
URL http://tools.ietf.org/html/rfc3225

[31] O. Kolkman and R. Gieben, "DNSSEC Operational Practices", RFC 4641, September 2006.
URL http://tools.ietf.org/html/rfc4641

[32] D. J. Bernstein, "Internet insecurity", University of Illinois at Chicago, October 2008.
URL http://cr.yp.to/talks/2008.10.10/slides.pdf

[33] D.J. Bernstein, "Curve25519: new Diffie-Hellman speed records", February 2006.
URL http://cr.yp.to/ecdh/curve25519-20060209.pdf

[34] IETF Tools, "draft-ietf-dnsext-ecc-key",Dnsext Status Pages, DNS Extensions (Active WG), March 2007. URL http://tools.ietf.org/wg/dnsext/draft-ietf-dnsext-ecc-key/

[35] NIST, "DNSSEC Frequently Asked Questions (and a few Frequently Heard Myths)", December 2008.
URL http://snad.ncsl.nist.gov/dnssec/faq.html

[36] R. Bellis and L. Phifer, "Test Report: DNSSEC Impact on Broadband Routers and Firewalls", Nominet, Core Competence, September 2008.
URL http://download.nominet.org.uk/dnssec-cpe/DNSSEC-CPE-Report.pdf

[37] D. Anderson, "My comment on NTIA docket number 0810021307–81308–01 (Enhancing the Security and Stability of the Internet's Domain Name and Addressing System)", AV8 Internet Inc.
URL http://www.ntia.doc.gov/DNS/comments/comment027.pdf

[38] P. Wouters, "Defending your DNS in a post-Kaminsky world", Xelerance, Black Hat Briefings 2009.
URL http://www.blackhat.com/presentations/bh-dc-09/Wouters/BlackHat-DC-09-Wouters-Post-Dan-Kaminsky-slides.pdf

[39] Jeremy Hitchcock, "ICANN 35: What's Going Down, Down Under (Want the Low Down?)", CircleID, June 2009.
URL http://www.circleid.com/posts/20090617_icann_35_whats_going_down_down_under/

[40] European Commission, “European Commission calls for an open, independent and accountable governance of the internet”, June 2009.
URL http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/951

[41] N4D: Networks for Development.
URL http://net4d.org/

[42] ICANN, “ICANN to Work with United States Government and VeriSign on Interim Solution to Core Internet Security Issue”, 3 June 2009.
URL http://www.icann.org/en/announcements/announcement-2-03jun09-en.htm

[43] SecSpider, A globally distributed polling system.
URL http://secspider.cs.ucla.edu/

[44] World Wide DNSSEC Deployment map.
URL http://www.xelerance.com/dnssec/

[45] DNSSEC-Logo.
URL http://www.dnssec-logo.org/

[46] ENISA, “Stock taking report on the technologies enhancing resilience of public communication networks in the EU member states”, 2009.
URL http://www.enisa.europa.eu/doc/pdf/resilience_tech_report.pdf

[47] DNSSEC.net, “DNSSEC Software, DNSSEC Tools, DNSSEC Utilities”.
URL http://www.dnssec.net/software

[48] DNSSEC Deployment Initiative.
URL http://www.dnssec-deployment.org/tracker/

[49] DNSSEC-tools, “Template:DNSSEC-Tools Components”, February 2008.
URL http://www.dnssec-tools.org/wiki/index.php/Template:DNSSEC-Tools_Components

[50] S. Krishnaswamy, W. Hardaker and R. Mundy, “Using DNSSEC-Tools to Deploy DNSSEC”, SPARTA Inc, in: Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security, 2009.
URL http://portal.acm.org/citation.cfm?id=1524292.1524310

[51] IANA, “Interim Trust Anchor Repository Beta”.
URL https://itar.iana.org/

[52] DNSSEC-Deployment, “STATEMENT OF NEEDED INTERNET CAPABILITY, Trust Anchor Repositories”, SPARTA Inc, Shinkuro Inc and National Institute of Science and Technology, June 2008.
URL http://www.dnssec-deployment.org/tar/tarpaper.pdf

[53] S. Weiler, “DNSSEC Lookaside Validation (DLV)”, RFC 5074, November 2007.
URL http://tools.ietf.org/html/rfc5074

[54] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei and W. Lee, “Increased DNS forgery resistance through 0x20-bit encoding security via leet queries”, in Proceedings of the 15th ACM conference on Computer and communications security, 2008.
URL http://portal.acm.org/citation.cfm?id=1455770.1455798

[55] P. Vixie and D. Dagon, “Use of Bit 0x20 in DNS Labels to Improve Transaction Identity”, Internet-Draft, March 2008.
URL http://tools.ietf.org/html/draft-vixie-dnsext-dns0x20-00

[56] A. Hubert and D. Ulevitch, “EDNS Option for performing a data PING draft-hubert-ulevitch-edns-ping-01.txt” Internet-Draft, April 2009.
URL http://www.ietf.org/internet-drafts/draft-hubert-ulevitch-edns-ping-01.txt

[57] W. Wijngaards, “Resolver side mitigations draft-wijngaards-dnsext-resolver-side-mitigation-01”, Internet-Draft, February 2009.
URL http://tools.ietf.org/id/draft-wijngaards-dnsext-resolver-side-mitigation-01.txt

[58] P. Vixie, O. Gudmundsson, D. Eastlake and B. Wellington, “Secret Key Transaction Authentication for DNS (TSIG)”, RFC 2845, May 2000.

URL http://tools.ietf.org/html/rfc2845

[59] S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead and R. Hall, “Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)”, RFC 3645, October 2003.
URL http://tools.ietf.org/html/rfc3645

[60] DNSTrust, “TLD dependency graph”, June 2009.
URL http://shinobi.dempsky.org/~matthew/dnstrust/graphs/

[61] E. Osterweil, V. Pappas, D. Massey and L. Zhang, “Zone State Revocation for DNSSEC”, in: Proceedings of the 2007 workshop on Large scale attack defense, 2007.
URL http://portal.acm.org/citation.cfm?id=1352664.1352677