Research Project 2

DNSsec Revisited

Anastasios Poulidis
Anastasios.Poulidis@os3.nl

Hoda Rohani
Hoda.Rohani@os3.nl

Supervisor: Jeroen Scheerder
Jeroen.Scheerder@on2it.net

System and Network Engineering
University of Amsterdam
July 11, 2014

Abstract

DNS is the naming system that translates domain names to IP addresses. DNSSEC is an extension of DNS that provides security to certain kinds of information that DNS provides. In this research project we present a measurement study that inspects the adoption rate of DNSsec among the most popular domains and evaluates the maintenance status that these domains have. For the adoption rate we confirmed that only a few domains have DNSsec-enabled in their zones, nearly 1%. For the maintenance status, our results showed that nearly 84% authoritative nameservers of signed domains return the same and consistent data while 8% return different data, consistent or inconsistent and the remaining 8% has only one active nameserver. Finally, from the DNSsec-enabled domains, our results showed that 3 out of 4 domains provide validated answers and the remaining do not provide validation due to misconfiguration or possible attacks.

Contents

1 Introduction

The Domain Name System (DNS) is the Internet naming service. Its functionality is mapping domain names to IP addresses and IP addresses to domain names [1]. Nowadays DNS becomes very crucial that almost all Internet applications are using it. So it is more necessary to secure and enhance it. Unfortunately, as the DNS protocol is very old, it has some vulnerabilities such as lack of authentication and data integrity. These weaknesses enable attackers to poison the caches of name servers and inject fake data. So what is the solution?

DNS Security Extensions or DNSsec can help us in these situations to reduce the compromise of the DNS infrastructure. DNSsec was created by IETF to meet the DNS requirements. Each zone creates public/private key pairs to sign its data. This brings us authentication and integrity checking. The private key is held by the administrator and kept secret while the public key is added to the zone in a resource record called a DNSKEY record.

Now the question is after over ten years of introducing DNSsec, how many domains are using it? and is it well-maintained or not? The existing statistics shows that DNSsec is not widely deployed while it provides valuable functionality. It seems that the complexity of deploying and maintaining the DNSsec hinders its usage. Generating key pairs and signing zones for the first time, resigning the entire zone after editing zone's records, and updating keys regularly before their expiration date for maintaining the DNSsec without any bugs are issues that administrators should be aware.

The structure of this paper is as follow: In Chapter 1 the research questions, our approach, the related work and the scope of our research are presented followed by Chapter 2 and Chapter 3 where DNS and DNSsec features and functionalities are described. In Chapter 4 the methodology that this research followed is defined. Finally in Chapter 5 the results of our experiments are presented followed by the Conclusion, in Chapter 6, and the Future Work, in Chapter 7, that can be made subsequently to our research.

1.1 Research Questions

The problem of proper management and the adoption rate we identified before, defines the following research questions:

Proposed Questions:

- What is the DNSsec adoption rate in the most popular domains?
- If the DNSsec is deployed in the zone, is it managed and operated properly?

    What are the causes of bogus DNSsec enabled zone?

1.2 Related Work

A lot of research has been done for DNSsec [2, 3, 4] while only a few of them were exploring the adoption rate [5, 6]. Internet Society Deploy360 Programme [1] presents a number of websites which show a collection of statistics about DNSsec for some specific gTLDs [2] and ccTLDs [3]. But all of them miss some parts of information mostly related to its management, such as checking signature validity or checking DS records in the parent zone. Here in this project we don't limit ourselves to specific TLDs and we are using the top popular domains to provide a view of the deployment of DNSsec, where it is really needed.

1.3 Scope

The scope of our research project is to investigate the DNS adoption rate and focus on the management of the most popular domains. Using DNSsec-aware resolvers by the users and the overall security options that the end users can adjust to benefit from DNSsec is out of the scope of this research project.

2 DNS

The Domain Name System (DNS) [7, 8] is a hierarchical distributed naming system used for mapping domain names to IP addresses. It is fundamental and required for every transaction on Internet like visiting a website or sending an email. With DNS a machine can locate other computers worldwide.

2.1 Domain Resolution

Each domain name consists of two or more sub-domain names separated by ".". The information that these sub-domains provide is distributed around the world in the hierarchical matter.

For example, a computer that needs to translate a domain name such as "os3.nl" will have to contact the two sub-domains that compose "os3.nl", "os3" and "nl" as well as the root "." which is the top level in the hierarchical structure and the location where the resolution starts from. Each domain including root, has its own IP addresses and each sub-domain in a lower level is depending on the domain lying in the upper level. As a result, the computer will have to contact the IP address of the root, which is known to the Operating System, to get the IP address of the sub-domain "nl". Then, it will have to contact the IP address of "nl" to get the IP address of "os3.nl". Finally, after resolving the whole domain, it will request the IP address related to the service queried and get access to it.

[1]See `http://www.internetsociety.org/deploy360/dnssec/statistics/`.

[2]See `http://fedv6-deployment.antd.nist.gov/`.

[3]See `https://xs.powerdns.com/dnssec-nl-graph/`.

There are 13 root servers, mirrored by many more around the world which can provide the IP address of the lower level domain. The top level domains consist of the gTLDs (com, net, org, gov) and the ccTLDs (nl, gr, uk) which can provide the IP address of the domains they have delegation for. Moreover, each administrator that registers a domain name, assigns IP addresses for its services (web, email) which are subsequently requested by the users at the last step.

Domain names are synonymous to DNS zones except that in the DNS zones administrative responsibility has been delegated to a single manager. As a result, a domain name can be partitioned in different sub-levels which are managed independently. A DNS zone is implemented in the configuration system of a domain name server in the form of a zone file containing all the required information needed for a domain to operate properly and provide the requested information.

DNS primarily uses User Datagram Protocol (UDP) on port number 53 to serve requests. DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server. The Transmission Control Protocol (TCP) is used in some cases when the response data size exceeds 512 bytes, or for tasks such as zone transfers.

2.2 DNS main components

DNS is built based on some basic architecture components needed from the time a user makes a query, which is propagated through these components to its final destination, all the way back to the time he receives his answer. These components are:

- Stub Resolver

  Stub resolver lies on the client side. It is responsible for initiating the queries that ultimately lead to a full resolution of a domain. It usually contacts and relies on a recursive resolver that will propagate the queries needed to resolve the requested domains and receive the answers for it.

- Recursive Resolver

  Recursive resolver is the intermediate component between the Stub Resolver belonging to the client side and the Authoritative Nameservers belonging to the server side where the domains are configured. As a result, it is the component that resolves all the sub-domains that compose the initial domain that client queried for. To achieve that, it contacts each sub-domain's nameserver starting from the root and ending to the final domain. Recursive resolver has the option to caches the answer it receives from each nameserver (Caching Resolver) for the time (TTL) provided by the nameserver. Using caches, improves efficiency, increases performance in end-user applications by returning faster answers and decreases the operating burden on the queried servers. ISP and large organisations provide recursive/caching resolvers to their users in case they have not configured one on their own.

- Authoritative Nameserver

  Each domain including root should contain at least two authoritative nameservers for better functionality and protection against cases of server failure and inability to respond to potential requests. An authoritative nameserver has the permission to return answers for each query comes to the server, related to a domain configured by the administrator. Each authoritative nameserver of a domain must return the consistent answer, in the order configured by the administrator. The return of the same answer from all authoritative nameservers of a domain is not always the case as we will present afterwards.

The DNS architecture and the resolution of the domain "os3.nl" that was discussed previously is presented here:

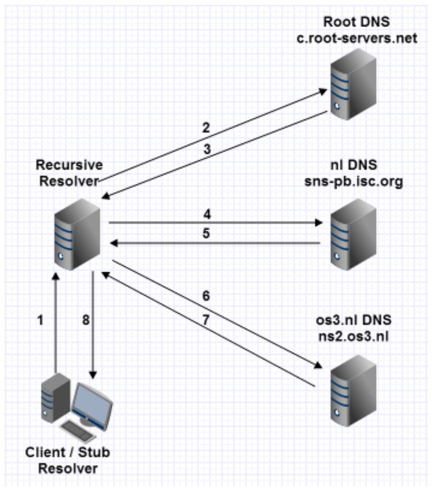


Figure 1: Domain Resolution

2.3 Resource Records

A resource record (RR) is the basic data element in the domain name system. It is contained in each domain zone and is returned as an answer to the DNS component that request it.

The attributes of a resource record are:

- Owner - Fully qualified domain name (FQDN) that this RR belongs.
- TTL - Time to live in the cache.
- Class - usually IN, Internet.
- Type - For what purpose it will be used such as A, AAAA, MX, CNAME, PTR, SOA, NS, DNAME and SRV.

  A - An A record defines an IPv4 address for a domain.

  AAAA - A quad-A record defines an IPv6 address for a domain.

CNAME - A canonical name record defines an alias for a domain.

DNAME - A DNAME record is used for non-terminal DNS Name Redirection.

MX - A Mail eXchanger record defines a mail server for a domain.

NS - A Name Server record defines a zone.

PTR - A PTR record is used for reverse lookup. Reverse lookup translates an IP to a domain.

SOA - Start Of Authority record administrates important zone parameters.

SRV - A service record specifies the location of the services that a domain supports.

- RData - Data related to the RR type.

The domain name system also supports wildcard DNS records which specify names that start with the asterisk label, "*". These type of records generate and match any query requesting any label followed by the FQDN of the zone. For example, if a domain with name "example.com" has an A record pointing to an IP address and uses wildcard record, when one will try to lookup any domain name ending in example.com, an A record will be generated for this domain that will point to the IP address that was configured. In addition to resource records defined in a zone file, the domain name system also defines several request types that are used only in communication with other DNS nodes (on the wire), such as when performing zone transfers (AXFR/IXFR) or for EDNS (OPT) [9].

2.4 Message Format

There are two types of DNS messages, queries and replies, and they both have the same format. Each message consists of a header and four sections each containing the related information followed by the TTL, class, type and data:

- Question: Contains the queried name.
- Answer: Contains the answer to the query.
- Authority: Contains the nameservers of the domain and other referral information .
- Additional: It usually contains data related to information contained in the Authority section.

The header section contains the fields:

Header Section	
ID	Identifier
Flags	QR - Query or Response OPCODE - Kind of Query AA - Authoritative Answer TC - Truncated Response RD - Recursion Desired RA - Recursion Available AD - Authentic Data (used by DNSsec) CD - Checking Disabled (used by DNSsec) RCODE - Result Code that indicates the behaviour of DNS
QDCOUNT	Number of questions
ANCOUNT	Number of answers
NSCOUNT	Number of authority records
ARCOUNT	Number of additional records

The most common result codes are:

Result Codes	
NOERROR	No error in the answer
SERVFAIL	Server failure
NXDOMAIN	The queried domain does not exist

The Flags and the result codes of a message header outline a big part of the overall behaviour of DNS and can be used as indication for troubleshooting. Flags play a big role especially in DNSsec (Domain Name System Security Extensions), where a part of the security of DNS can be evaluated through them. Authoritative answers are the responses come from an authoritative nameserver. These queries cannot provide useful information about the overall security of the service because they offer a dimensional informationa and not a multidimensional from the root down to the queried domain as required in DNSsec. Authentic Data provide one of the most useful security related information indicating that the security of the service is the desired one or not. Checking disabled is used primarily for troubleshooting purposes in combination with the ‘SERVFAIL’ result code which together can indicate the security problem behind the deployment of DNSsec.

2.5 DNS Vulnerabilities

The DNS was not built with security in mind and as result has many vulnerabilities including forged answers, DDoS attacks, amplification attacks, interception and modification of response packets, fake or manipulated name servers of a zone, cache poisoning, registry compromise, bogus routes and others. Most cyber attacks are related to cache poisoning and DDoS attacks.

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is focused on making a computer resource unavailable to its intended users. A DDoS consists of the concerted efforts to prevent an Internet site or service from functioning efficiently or at all. A possible (D)DoS attack to a DNS server will make it unavailable to return the IP address needed to access the web or email server.

A DNS Amplification attack is a popular form of DDoS, in which attackers use publically accessible open DNS servers to flood a target system with DNS response traffic.

Cache poisoning is one of the most dangerous and common attacks on DNS. An attacker that make use of this attack, by injecting bogus data in the cache of a DNS resolver making it respond with false data to queries made by the user. In this way an attacker can inject a malicious mapping between a name and an IP, redirecting the user to a malicious web or email server.

DNS does not provide any encryption and as a result every packet is travelling in clear text through the wire. An attacker can spoof possible response packets and modify them accordingly.

A domain name registrar is an entity that manages the reservation of Internet domain names. One attacker can gain unauthorised access to registrar account and change the victim zone's delegation to point at bogus name servers. Moreover, a zone can be target of an attack when an attacker takes control of the nameserver of a zone and returns false data or one can create a fake name servers for a zone and trick other nameservers to query the fake one.

Finally, an attacker can guess possible outgoing queries of a resolving server and try to answer with forged data or brute force possible answers.

The latter DNS vulnerabilities make a DNS client unsure about some important questions related to the security of the service, such as, where an answer really came from, if the server replied is telling the truth or not and if it received exactly what the server sent among others.

3 DNSsec

The Domain Name System Security Extensions (DNSSEC) [10, 11, 12, 13, 14] is an Internet service that enhances DNS with a few security features. It provides DNS clients and resolvers with origin authentication of DNS data, authenticated denial of existence, data integrity, but not availability or confidentiality.

All these benefits that DNSsec provides, comes from the signing of the information that DNS zones contain. As a result, a DNS resolver that receives the answer, knows the authenticated origin of the DNS information and can verify it by contacting the authoritative nameserver and evaluate the integrity of its data. In this way, it also authenticates who signed the data, provides non-repudiation and protection of the resolvers from forged or manipulated DNS data created by cache poisoning. Moreover, DNSsec provides a secure bonding between child and parent zones (chain of trust). By doing this, it protects nameservers from possible attacks and manipulation of their data or use of fake nameservers. As a result, the parent zone can verify the integrity of child zone's data and notify the user about it.

DNSsec does not provide protection against (D)DoS attacks and manipulation of the data as they travel through the wire because DNS data are public and they travel in clear text resulting in possible spoofing. Other technologies or mechanisms can protect from these effects such as IPSec, TSIG [15] or SIG [16] but not in all aspects of DNS and without increasing its complexity.

3.1 DNSsec characteristics

DNSsec introduces some new attributes and records needed in order to provide the desired security. It offers public-key cryptography and applies it in the records of a zone by digitally signing them. The correct DNSKEY record is authenticated via a chain of trust, starting with a set of verified public keys for the DNS root zone which is the trusted third party and ending with the domain zone. The proper authentication is possible only if the domain administrator has uploaded the hash of the DNSKEY (KSK) to the parent zone.

The Records that DNSsec introduces are:

RRSIG	contains the DNSsec signature for a record set.
DNSKEY	contains the public key that a DNS resolver uses to verify DNSsec on the RRSIG of the record sets.
DS	It is placed in the parent zone, references a sub-delegated zone (child) and is used to verify the validity of DNSsec by containing the hash of the DNSKEY of the delegated zone.
NSEC	Contains a link to the next record name in the zone and lists the record types that exist for the record's name.
NSEC3	It is a more secure version of NSEC providing the next record name in the zone in hashed name sorting order.
NSEC3PARAM	Contains parameters of NSEC3.

3.2 DNSsec Verification procedure

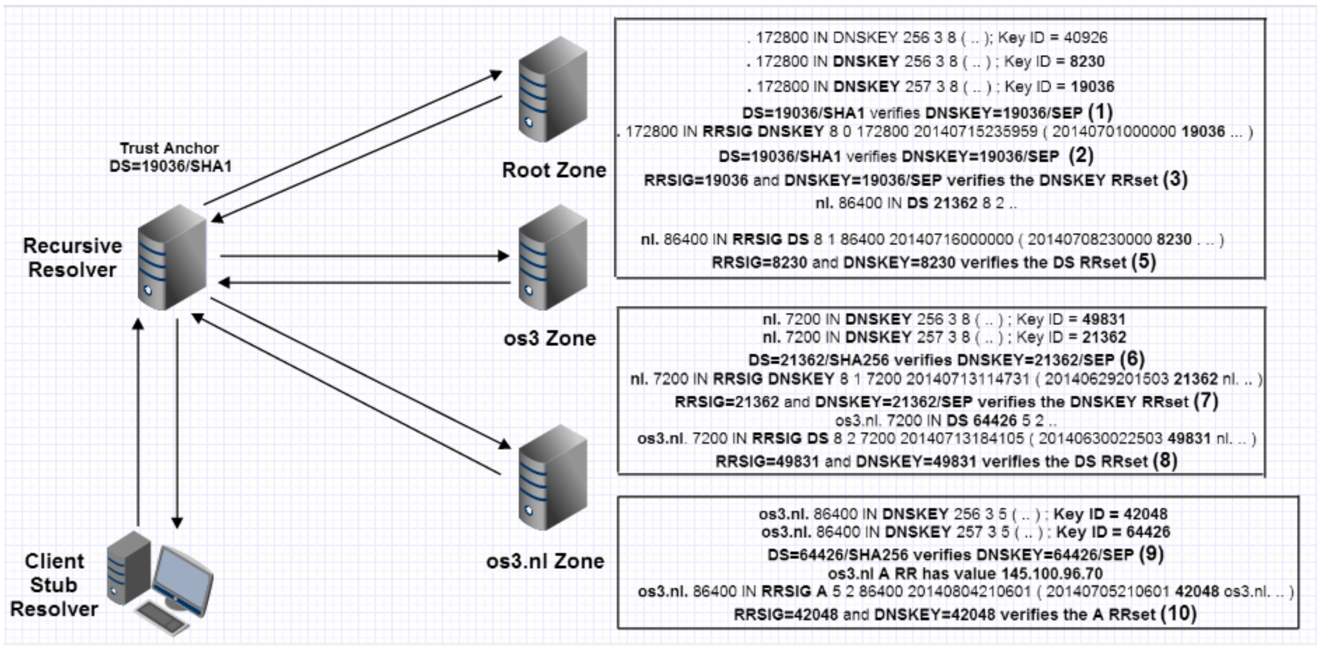
When DNSsec is configured, the resolving of a domain name is followed by the verification of the requested record signature and the proper delegation starting with the root and ending with the requested domain.

When the client that uses DNSsec-aware resolvers tries to lookup a domain name that is DNSsec enabled, makes use of a recursive query mode by his Stub resolver, which forwards the request to a recursive name server. The recursive name server uses the chain of trust model and starts making a query to a Trust anchor, usually the root. Eventually, an answer to a DNSsec resolution can be authenticated/validated by checking the authentication of the complete chain.

Using the chain of trust model, a Delegation Signer (DS) record in a parent domain (DNS zone) can be used to verify a DNSKEY record in a sub-domain, which can then contain other DS records to verify further sub-domains.

The recursive nameserver begins validation by verifying the DS and DNSKEY records at the DNS root. Then it uses the DS records for the top level domain "tld" found at the root to contact and verify the DNSKEY records in the "tld" zone. Subsequently, it checks if there is a DS record for the "domain.tl" sub-domain in the "tld" zone, and if there is one, it will contact the sub-domain and use the DS record to verify the DNSKEY record found in the "domain.tld" zone. Furthermore, it will request the answers for the A records found in "domain.tld" and verify their RRSIG records. Finally, the recursive resolver will return the answer to the stub resolver with an Authenticated Data (AD) flag, indicating the proper validation of the data. In our example, "domain" can be any valid domain or sub-domain and "tld" can be any gTLD or ccTLD, such as "com", ".nl", ".gr". We also assumed that the domain is not cached at the resolver.

An example is illustrated below:



```
. 172800 IN DNSKEY 256 3 8 ( .. ); Key ID = 40926
. 172800 IN DNSKEY 256 3 8 ( .. ) ; Key ID = 8230
. 172800 IN DNSKEY 257 3 8 ( .. ) ; Key ID = 19036
DS=19036/SHA1 verifies DNSKEY=19036/SEP (1)
. 172800 IN RRSIG DNSKEY 8 0 172800 20140715235959 ( 20140701000000 19036 ... )
DS=19036/SHA1 verifies DNSKEY=19036/SEP (2)
RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset (3)
nl. 86400 IN DS 21362 8 2 ..

nl. 86400 IN RRSIG DS 8 1 86400 20140716000000 ( 20140708230000 8230 . .. )
RRSIG=8230 and DNSKEY=8230 verifies the DS RRset (5)
```

```
nl. 7200 IN DNSKEY 256 3 8 ( .. ) ; Key ID = 49831
nl. 7200 IN DNSKEY 257 3 8 ( .. ) ; Key ID = 21362
DS=21362/SHA256 verifies DNSKEY=21362/SEP (6)
nl. 7200 IN RRSIG DNSKEY 8 1 7200 20140713114731 ( 20140629201503 21362 nl. .. )
RRSIG=21362 and DNSKEY=21362/SEP verifies the DNSKEY RRset (7)
os3.nl. 7200 IN DS 64426 5 2 ..
os3.nl. 7200 IN RRSIG DS 8 2 7200 20140713184105 ( 20140630022503 49831 nl. .. )
RRSIG=49831 and DNSKEY=49831 verifies the DS RRset (8)
```

```
os3.nl. 86400 IN DNSKEY 256 3 5 ( .. ) ; Key ID = 42048
os3.nl. 86400 IN DNSKEY 257 3 5 ( .. ) ; Key ID = 64426
DS=64426/SHA256 verifies DNSKEY=64426/SEP (9)
os3.nl A RR has value 145.100.96.70
os3.nl. 86400 IN RRSIG A 5 2 86400 20140804210601 ( 20140705210601 42048 os3.nl. .. )
RRSIG=42048 and DNSKEY=42048 verifies the A RRset (10)
```

Figure 2: Chain of Trust

When a resolver is configured to validate the data, three possible answers are available.

- Secure data accompanied with NOERROR status and AD flag
- Insecure data with NOERROR status without AD flag
- Bogus data with SERVFAIL status

Secure data exist when there is an unbroken chain of trust from the trust anchor to the RRset. If the resolver cannot follow the chain of trust but can securely show that no such chain exists, the result is insecure data. When there is no link between parent and child, the parent must prove that there is no DS record for the child zone. The authenticated denial of existence is accomplished by NSEC records. The answer is bogus when the resolver cannot track the chain of trust and prove that no such chain exists. Figure 3 shows these three situations.

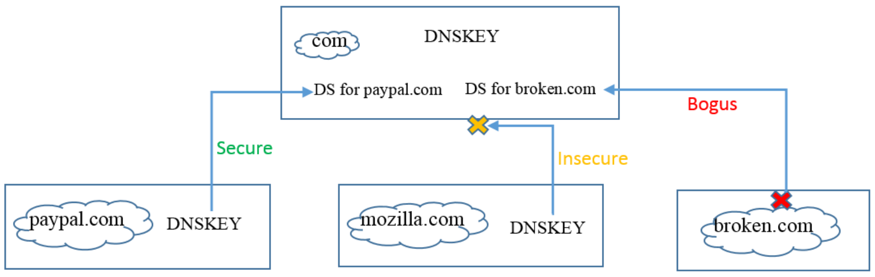


Figure 3: Secure-Insecure-Bogus

In the situation of a "SERVFAIL" a validating stub resolver can perform its own signature validation by setting the Checking Disabled (CD) bit in its query messages. A validating stub resolver uses the CD bit to perform its own recursive authentication and can evaluate if the failed validation is due to bogus authentication chain, configuration error, network failure or possible attack.

3.3 Weaknesses and overhead of DNSsec

Although the DNSsec protocol provides advanced security solutions to the DNS protocol, it is still vulnerable to some types of attacks as well as introduces a computational overhead.

The administrator that implements DNSsec in the domain's zone has to perform some additional operations such as generating keys, signing the zone, publishing the hash of the key to the parent zone and the general maintenance of an additional service in the DNS server. Any possible mistakes or unawareness in the DNSsec configuration and maintenance can lead to unavailability for both DNS and DNSsec services.

In addition, with the signing of the records in the zone, there is an additional need for Memory and CPU utilization, as a result of the larger DNSsec answers and zone size that need to be processed. The size of DNSsec zone file becomes seven times larger than an unsigned file. This has an impact both for the server side but also for the query side.

DNSsec also increases the DNS packet size, making resolvers and nameserver to need more bandwidth to send and receive these large packets and to introduce additional network overhead related to PMTUD (Path MTU Discovery), fragmentation of the DNS packets and possible interference by firewalls and proxies and middle-boxes having difficulties handling these packets.

These large responses can be misused for DOS amplification attacks to clog victims' networks and hosts.

3.4 Key management

Besides the common TTL values that are used for caching purposes, DNSsec uses additional timestamps in the RRSIG records, for protection against replay attacks and attempts to retrieve the DNSKEY. These timestamps are attached when the DNSKEY is generated and is used to sign the domain zone and records that contains in it. As a result, these timestamps must be refreshed often, by re-signing the zone and re-distributing them to the delegated servers. Neglecting key rollovers, will result in 'SERVFAIL' by the validating resolvers.

The key rollover scheme is a complicated and expensive task containing many steps that must be performed in the correct order and without missing any. In addition, when the administrator generates the DNSKEY, at first, generates the Key Signing Key (KSK) which is used to generate and sign the other DNSKEY records, which are usually the Zone Signing Keys. The Zone Signing Keys (ZSK) are used to sign other resource records and eventually the whole zone. The hash of KSK is transferred and published in the parent zone to provide delegation and validation of the child zone. KSK is more important than ZSK and also needs to be switched less often. As it is larger, it is harder to crack while it needs more complicated steps in order to rollover that includes participation of more than one zone. Instead, the ZSK is used only in one zone, thus, can be much shorter and be switched more easily.

ZSK and KSK rollover schemes use different methods that share both similarities and differences.

ZSK rollover uses the Pre-publish method and the required steps are:

1. Generate new ZSK, add key to zone (Increase zone's serial number)
2. Re-sign zone with using old key and KSK
3. Wait for the TTL of the zone to pass
4. Re-sign with the new key but leave the old zsk published in the zone
5. After all records signed with the old private key have expired (wait zone propagation time + largest TTL of all records in the zone), remove old key
6. Resign one last time

KSK rollover uses the Double Signature method and its steps are:

1. Generate new KSK, add new KSK to the zone and sign the DNSKEY RRset with both keys
2. Wait for the TTL of the zone to pass
3. Upload new DS to the parent zone
4. When new DS RR appears in the zone, wait TTL of the old DS record

5. Remove the old KSK and resign zone
6. Remove old DS record from parent

There are multiple algorithms for KSK and ZSK which provide different hashing and encryption mechanisms that DNSsec makes use to protect the domains. These algorithm are:

Algorithm	Status
RSA/MD5	Deprecated
DSA/SHA-1	Optional
RSA/SHA-1	Required
RSASHA1-NSEC3-SHA1	Recommended
RSA/SHA-256 RFC 5702	Recommended
RSA/SHA-512	Recommended
GOST R 34.10-2001	Optional
ECDSA/SHA-256	Recommended
ECDSA/SHA-384	Recommended

With the exception of RSA/MDA which is deprecated any other algorithm can be used freely. The default algorithm used for signing is RSA/SHA-1.

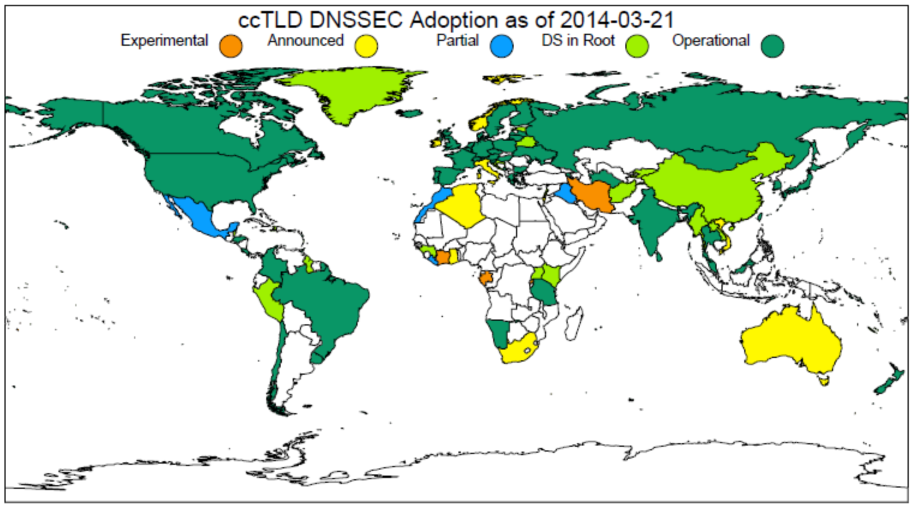
3.5 DNSsec deployment

The root trust anchor is the first component that is queried to begin validating a DNSsec zone, followed by the gTLDs and ccTLDs and ended with the domains and subdomains of the requested zone. If any intermediate component is not configured correctly or lacks DNSsec the validation fails. The components that are the most important in DNSsec deployment are root and TLDs zones, in which a lot of zones rely on their security to provide validation to their end users. Root and TLDs was not DNSsec enabled from the start of DNSsec, but the last 4 years more and more TLDs are being signed offering the DNSsec security mechanisms that DNSsec was build for. On July 15 2010, DNSsec was first deployed at the root level, with the ".org" top-level domain being signed a month earlier in June 2010. More gTLDs and ccTLDs followed, including .com, .net, and .edu later in 2010 and 2011. Nowadays, 636 TLDs are registered in the root zone in total, with 451 of them being signed, leading to a 71% DNSsec signed TLDs. Also, 441 of them have trust anchors published DS records in the root zone and 5 TLDs have trust anchors published in the ISC DLV Repository, offering the desired validation.

A worldwide map presenting the DNSsec deployment status is presented here:

4 Methodology

Initially, we had to find a large set of domains where we could apply our experiments of adoption rate and maintenance status. We found this set in the Alexa website where the top 1 million



Experimental -- Internal experimentaion announced or observed (5):	CI GA IR RW TO
Announced -- Public commitment to deploy (11):	AU DZ GH HK IE IL IT NO SG VN ZA
Partial -- Zone is signed but not in operation (no DS in root) (6):	IQ LR MA MS MX VC
DS in Root -- Zone is signed and its DS has been published (27):	AF AG BY BZ CC CN EE FO GI GL GN GY HR HT KE KG KI LA LB LC MM NC NU PE PW TV UG
Operational -- Accepting signed delegations and DS in root (65):	AC AM AT BE BG BR CA CH CL CO CR CX CZ DE DK ES FI FR GR GS HN HU IN IO IS JP KR LI LK LT LU LV ME MN MU MY NA NF NL NZ PL PM PR PT RE RU SB SC SE SH SI SX TF TH TL TM TT TW TZ UA UK US UY WF YT

Figure 4: TLD DNSsec adoption

websites are stored and updated daily. This set provides websites and not domains and as a result we had to extract only the unique domains and remove any duplicate domains, URL paths, invalid URLs and IP addresses. The final set of unique domains was counted approximately to 930000.

Afterwards, we used dig command-line tool to query these domains and by using the "+dnssec" option we could verify the validation that DNSsec provides for these domains that have implemented it. When the "+dnssec" option is used by a DNSsec-aware resolver, the "DO" flag bit is set in the DNS query. Flag bit "DO" is in the extended flag bits defined by EDNS (Extension mechanisms for DNS) and all the intermediate machines must support EDNS and DNSsec. EDNS supports also large packets that are essential for proper DNSsec functionality.

From this initial experiment we could realise, how many domains are DNSsec enabled by checking the existence of RRSIG, but occasionally without the desired accuracy. The first observation that confirmed that was the missing in some cases of an answer by the domains nameserver. This was a result of the use of the default query that requests A records explicitly and in cases of lack of A records in the configured nameserver side it could not return any answers. In order to bypass this situation we used "+any" option in our queries. The use of "+any" flag provide us with all the records of the zone, where we had to filter the ones that are not needed such as NSEC, which is related to non-existence domains and DS, which is related to the delegation of the parent zone.

When a domain is queried, the authoritative nameserver of the zone responds to the query with the requested information. Most domains are configured with more than one authoritative nameservers and depending on the administrator's setup, everyone can respond to the request. In some cases, the nameservers lay on different domains making the things more complicated and introducing configuration errors and negligence by the nameservers' administrators. As a result different nameservers for the same domain can return different answers making our results inconsistent.

As a result in our next step, we gathered all the authoritative nameservers of the domains and we queried them to verify the existence or lack of DNSsec by confirming the appearance of RRSIGs. By doing this, we could confirm the consistency of the nameservers' answers but without having the option to check the validity of their responses.

Subsequently, we queried the domains that had consistent nameservers and we verified the existence or lack of validated answers by confirming the existence of RRSIG records and AD flags. In addition, we checked the domains containing inconsistent nameservers to verify the existence of DNSsec in any nameserver, that would imply that DNSsec is configured in the domain but not effectively to all nameservers.

When a query with "+dnssec" option is being made, the most common result codes are: "NOERROR", "SERVFAIL" and "NXDOMAIN". The "NOERROR" code indicates that the resolution was successful and the results are returned with validation or non-validation. The "NXDOMAIN" code indicates that the domain does not exist and an NSEC record is returned as an answer with validation or non-validation in case of a domain that supports DNSsec.

In case of a "SERVFAIL" code, a problem to the network, server, a configuration error or a possible attack is implied. In order to distinguish these situations the "CD" flag was used, that attempts to lookup the domain without requesting validation. For the nameservers and domains that returned the "ServFail" code, we repeated our experiments to distinguish the situations where DNSsec, the server/network or a possible attack caused the failure.

In these situations, where we had to repeat our experiments multiple times using two Google Public DNS resolvers and our server to be sure that all the potential answers from different nameservers were seen.

Finally, after collecting the total DNSsec-enabled domains, we could perform some additional tests including:

- Validation Errors
- Cryptographic algorithms used for DNSKEY
- RRSIG lifetime
- (Non)Existence of NSEC/NSEC3 records

The overall lookup procedure that was used is presented below:

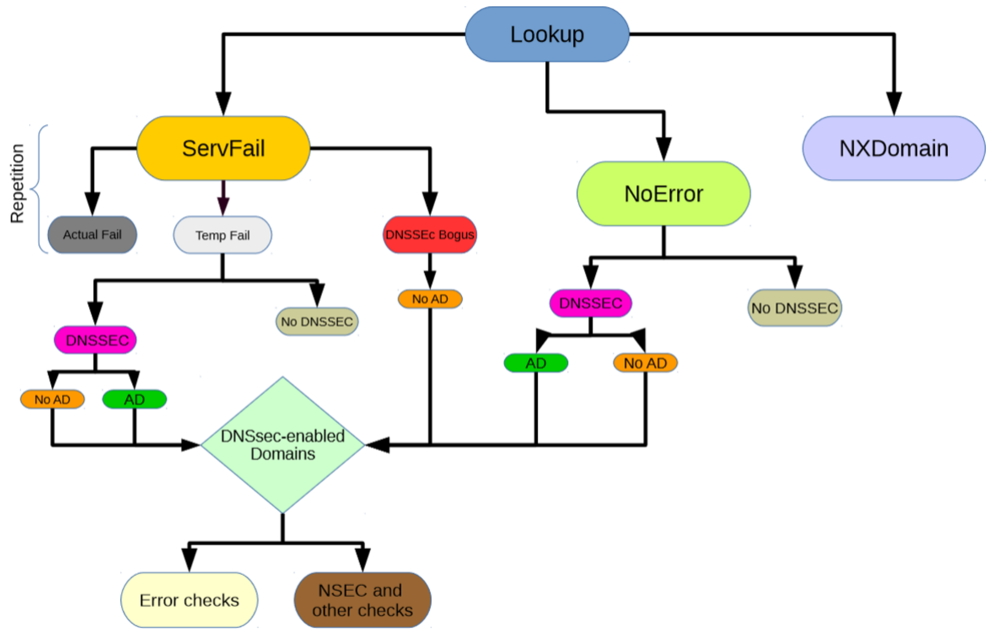


Figure 5: Lookup procedure

5 Results

In this section the results of our experiments are presented. The experiments that was performed were related to the DNSsec adoption rate among the top 1 million domains and the maintenance responsibility that they had maintained. Subsequently, the consistency of the answers that the authoritative nameservers returned was measured and the configuration errors that zone administrators do, was also investigated. Finally, experiments related to best practices for DNSsec deployment in the existing DNSsec-enabled domains were performed.

5.1 Nameservers

From our results, we found out that the answer returned by validating resolver is directly depending on which nameserver is used by the resolver. In some cases, the data in different nameservers of specific domain is different and even inconsistent with each other, with some of them providing validation and with others not providing validation or not returning signed records at all, that can lead to MiTM and malicious mapping by an attacker.

Here is our observations from our result:

- on average each domain has 3.2 nameservers.
- 772 domains have only one active nameserver (8.1%), that can lead to inability in answering the requests if the nameserver goes down.
- Nearly 84% of signed domains have multiple nameservers with exactly the same data in all of them.

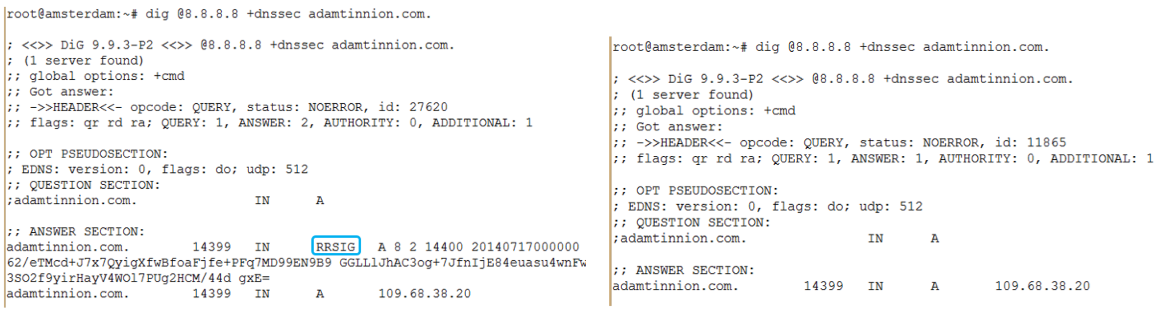
- Nearly 8% of signed domains have multiple nameservers with different data which is consistent or inconsistent with each other.

5.1.1 Inconsistencies

As part of our investigation, we tried to find out the inconsistencies in nameservers and categorise them accordingly.

- RRSIG and no RRSIG: We observed that some nameservers belonging to a specific domain have signed the zone, while the other nameservers have not signed it.

  Let's assume the "adamtinnion.com" domain. We requested that domain twice by a validating resolver. As you can see in Figure 6 the first answer includes RRSIG data even with AD flag which means that it is valid while the second one does not include any.



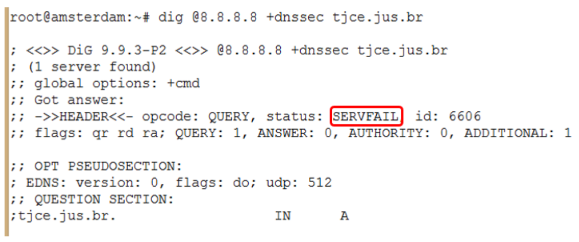
```
root@amsterdam:~# dig @8.8.8.8 +dnssec adamtinnion.com.

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec adamtinnion.com.
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27620
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;adamtinnion.com.              IN      A

;; ANSWER SECTION:
adamtinnion.com.        14399   IN      RRSIG   A 8 2 14400 20140717000000
62/eTMcd+J7x7QyigXfwBfoaFjfe+PFq7MD99EN9B9 GGLLlJhAC3og+7JfnIjE84euasu4wnFw
3SO2f9yirHayV4WOl7PUg2HCM/44d gxE=
adamtinnion.com.        14399   IN      A       109.68.38.20
```

```
root@amsterdam:~# dig @8.8.8.8 +dnssec adamtinnion.com.

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec adamtinnion.com.
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11865
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;adamtinnion.com.              IN      A

;; ANSWER SECTION:
adamtinnion.com.        14399   IN      A       109.68.38.20
```

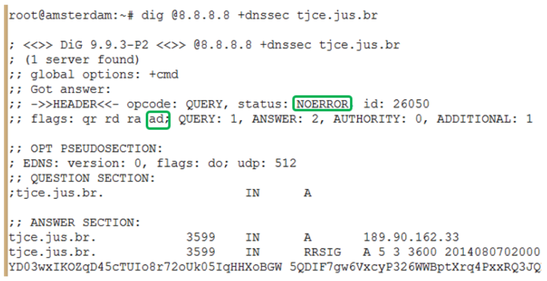
Figure 6: Inconsistencies in the nameservers

  We found 235 domains in this category of inconsistent nameservers.

- SERVFAIL and NOERROR: The inconsistencies in nameservers do not limit to RRSIG data. Consider tjce.jus.br domain in Figure 7, the first answer is SERVFAIL and the second answer is NOERROR. After repeating the experiment and using "+cdflag", we became sure that the SERVFAIL is due to bogus DNSsec, therefore, we realised that at least one of the nameservers of "tjce.jus.br" domain has DNSsec problem.



```
root@amsterdam:~# dig @8.8.8.8 +dnssec tjce.jus.br

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec tjce.jus.br
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 6606
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;tjce.jus.br.                   IN      A
```



```
root@amsterdam:~# dig @8.8.8.8 +dnssec tjce.jus.br

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec tjce.jus.br
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26050
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;tjce.jus.br.                   IN      A

;; ANSWER SECTION:
tjce.jus.br.            3599    IN      A       189.90.162.33
tjce.jus.br.            3599    IN      RRSIG   A 5 3 3600 2014080702000
YD03wxIKOZqD45cTUIo8r72oUk05IqHHXoBGW 5QDIF7gw6VxcyP326WWBptXrq4PxxRQ3JQ
```

Figure 7: tjce.jus.br - Inconsistencies in its nameservers

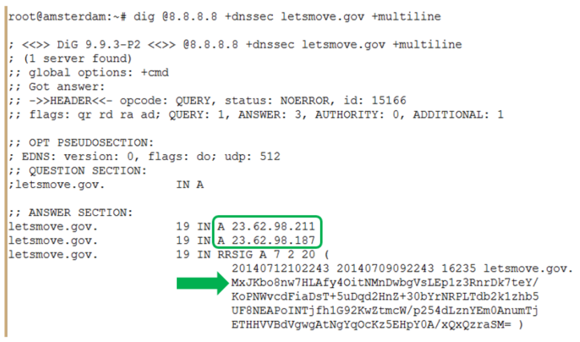
As you can see, these domains have non-deterministic behaviour when asked by recursive resolvers, so we label those domains as faulty non-deterministic ones.

Inconsistencies among authoritative servers can be due to an outdated or incorrect version of zone data served by the server or inconsistency in the level of DNSsec support. So for a proper DNSsec deployment, careful coordination and monitoring both vertically (between zone and its ancestor) and horizontally (between authoritative servers for the same zone) is needed.

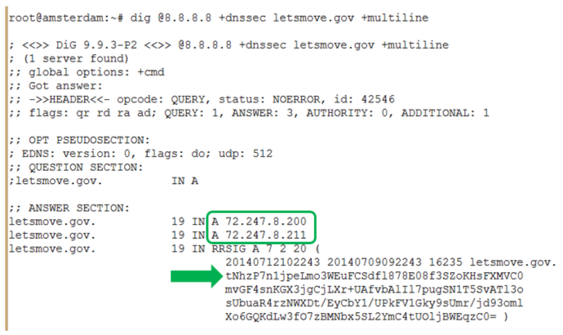
5.1.2 Different but consistent data in nameservers

It is worth mentioning that not all different data in nameservers cause failures, as long as the data is in consistent with each other and they provide validation on their answers, there is no problem. The fault arises when we have different inconsistent data.

A records: In some cases, there are multiple A records for one domain, and different nameserver serve different RRset of A for that domain. In Figure 8 both answers have different A records with different RRSIG. Note that all of these A records are valid and can be used (both have AD flag).



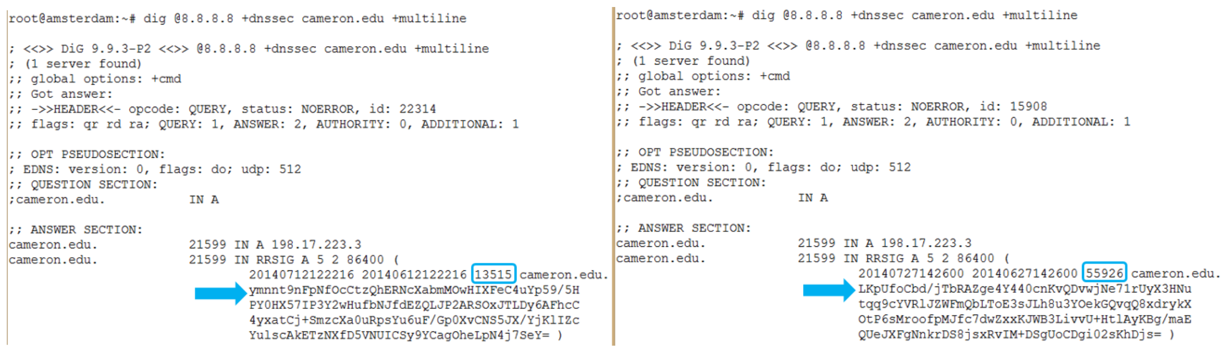
```
root@amsterdam:~# dig @8.8.8.8 +dnssec letsmove.gov +multiline

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec letsmove.gov +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15166
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;letsmove.gov.          IN A

;; ANSWER SECTION:
letsmove.gov.           19 IN A 23.62.98.211
letsmove.gov.           19 IN A 23.62.98.187
letsmove.gov.           19 IN RRSIG A 7 2 20 (
                                20140712102243 20140709092243 16235 letsmove.gov.
                                MxJKbo8nw7HLAfy4OitNMnDwbgVsLEp1z3RnrDk7teY/
                                KoPNWvcdFiaDsT+5uDqd2HnZ+30bYrNRPLTdb2k1zhb5
                                UF8NEAPoINTjfh1G92KwZtmcW/p254dLznYEm0AnumTj
                                ETHHVVBdVgwgAtNgYqOcKz5EHpY0A/xQxQzraSM= )
```



```
root@amsterdam:~# dig @8.8.8.8 +dnssec letsmove.gov +multiline

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec letsmove.gov +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42546
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;letsmove.gov.          IN A

;; ANSWER SECTION:
letsmove.gov.           19 IN A 72.247.8.200
letsmove.gov.           19 IN A 72.247.8.211
letsmove.gov.           19 IN RRSIG A 7 2 20 (
                                20140712102243 20140709092243 16235 letsmove.gov.
                                tNhzP7n1jpeLmo3WEuFCSdf1878E08f3SZoKHsFXMVC0
                                mvGF4snKGX3jgCjLXr+UAfvbAlI17pugSN1T5SvAT13o
                                sUbuaR4rzNWXDt/EyCbY1/UPkFV1Gky9sUmr/jd93oml
                                Xo6GQKdLw3fO7zBMNbx5SL2YmC4tUOljBWEqzC0= )
```

Figure 8: letsmove.gov - nameservers

Multiple ZSKs: Sometimes, nameservers have multiple Zone Signing Keys and different nameserver might use different ZSK to sign its zone. As a result, different nameservers return different signatures for the same data.



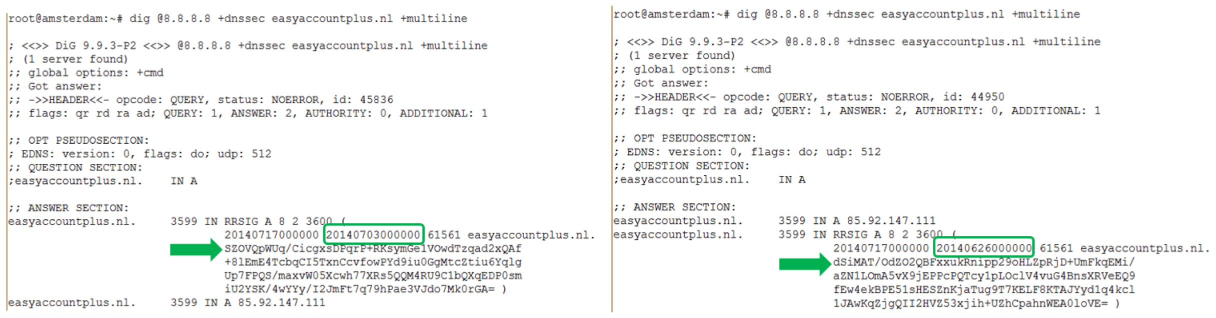
```
root@amsterdam:~# dig @8.8.8.8 +dnssec cameron.edu +multiline

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec cameron.edu +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22314
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;cameron.edu.           IN A

;; ANSWER SECTION:
cameron.edu.            21599 IN A 198.17.223.3
cameron.edu.            21599 IN RRSIG A 5 2 86400 (
                                20140712122216 20140612122216 13515 cameron.edu.
                                ymnnt9nFpNfOcCtzQhERNcXabmMOwHIXFeC4uYp59/5H
                                PY0HX57IP3Y2wHufbNJfdEZQLJP2ARSOxJTLDy6AFhcC
                                4yxatCj+SmzcXa0uRpsYu6uF/Gp0XvCNS5JX/YjKlIZc
                                YulscAkETzNXfD5VNUICSy9YCagOheLpN4j7SeY= )
```

```
root@amsterdam:~# dig @8.8.8.8 +dnssec cameron.edu +multiline

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec cameron.edu +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15908
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;cameron.edu.           IN A

;; ANSWER SECTION:
cameron.edu.            21599 IN A 198.17.223.3
cameron.edu.            21599 IN RRSIG A 5 2 86400 (
                                20140727142600 20140627142600 55926 cameron.edu.
                                LKpUfoCbd/jTbRAZge4Y440cnKvQDvwjNe71rUyX3HNu
                                tqq9cYVRlJZWFmQbLToE3sJLh8u3YOekGQvqQ8xdrykX
                                OtP6sMroofpMJfc7dwZxxKJWB3LivvU+HtlAyKBg/maE
                                QUeJXFgNnkrDS8jsxRvIM+DSgUoCDgi02sKhDjs= )
```

Figure 9: cameron.edu

The zones belonging to "cameron.edu" are signed by at least two ZSK which result in different returned RRSIG as shown in Figure 9.

Different Time: In Figure 10 we have two different RRSIGs signed by the same ZSK with different inception time. Both answers are valid and have AD flag. We found 475 domains with multiple nameservers having difference timestamps in their RRSIGs.



```
root@amsterdam:~# dig @8.8.8.8 +dnssec easyaccountplus.nl +multiline

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec easyaccountplus.nl +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45836
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;easyaccountplus.nl.    IN A

;; ANSWER SECTION:
easyaccountplus.nl.     3599 IN RRSIG A 8 2 3600 (
                                20140717000000 20140703000000 61561 easyaccountplus.nl.
                                SZOVQpWUq/CicgxsDPqrP+RKsymGelVOwdTzqad2xQAf
                                +8lEmE4TcbqCI5TxnCcvfowPYd9iu0GgMtcZtiu6Yqlg
                                Up7FPQS/maxvW05Xcwh77XRs5QQM4RU9C1bQXqEDP0sm
                                iU2YSK/4wYYy/I2JmFt7q79hPae3VJdo7Mk0rGA= )
easyaccountplus.nl.     3599 IN A 85.92.147.111
```

```
root@amsterdam:~# dig @8.8.8.8 +dnssec easyaccountplus.nl +multiline

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec easyaccountplus.nl +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44950
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;easyaccountplus.nl.    IN A

;; ANSWER SECTION:
easyaccountplus.nl.     3599 IN A 85.92.147.111
easyaccountplus.nl.     3599 IN RRSIG A 8 2 3600 (
                                20140717000000 20140626000000 61561 easyaccountplus.nl.
                                dSiMAT/OdZO2QBFxxukRnipp29oHLZpRjD+UmFkqEMi/
                                aZN1LOmA5vX9jEPPcPQTcy1pLOclV4vuG4BnsXRVeEQ9
                                fEw4ekBPE51sHESZnKjaTug9T7KELF8KTAJYyd1q4kcl
                                1JAwKqZjgQII2HVZ53xjih+UZhCpahnWEA0loVE= )
```

Figure 10: easyaccountplus.nl

Difference in SOA and RRSIG: Normally, before re-signing the zone, the administrator increases the serial number of SOA. This informs secondary servers to know that there is an update of the zone available, by comparing their zone's serial number with master's zone. Note that the dnssec-signzone utility by default doesn't increment the serial number itself and should be used carefully.

On the one hand, if the difference between the two serial numbers is small, some time for propagating the changes might be needed. On the other hand, if the difference is big, then there are definitely some errors.

As we discussed in this section, it is critical for the administrator of a zone to be sure that all authoritative servers have the most current version of the zone. The signed zone requires refreshing to prevent expired data and invalid DNSKEY. In DNSsec when the signatures are expired the whole zone will be invalidated.

5.2 DNSsec deployment status

For this experiment and for a 4-week time period, we queried the top 930000 unique domains to find out how many have deployed DNSsec. Furthermore, by having already obtained and categorised the domains that had deterministic and non-deterministic behaviour, we investigated the deterministic ones to figure out how many domains return validated or non-validated answers. The results showed that on average only 9916 (1%) of the zones are DNSsec-enabled which imply that DNSsec deployment is still relatively low. From the DNSsec-enabled zones, on average 7562 (76%) returned validated answers and 2355 (24%) returned not validated answers. The graph of our experiments is presented in Figure 11.

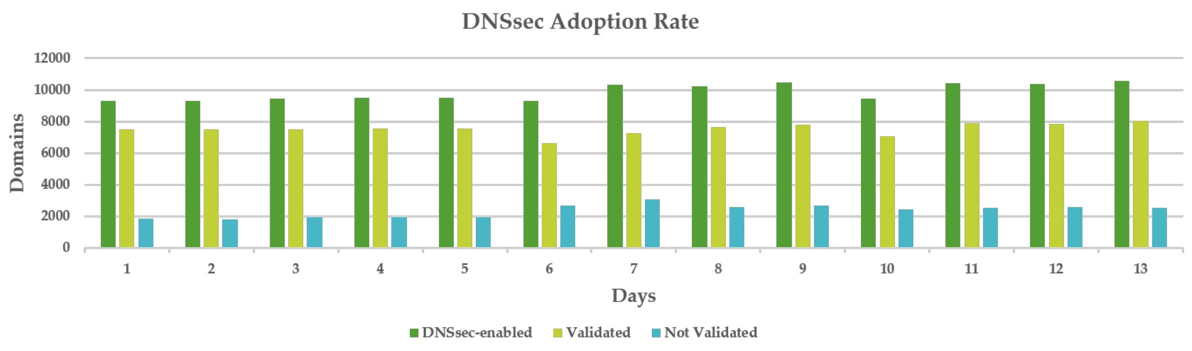


Figure 11: Adoption rate

After investigating the signed domains that do not return validated answers in order to find out the reason behind it, we observed that all of them have no DS record in the parents' zones. In Figure 12, two example of domains with AD and without AD flag are presented.

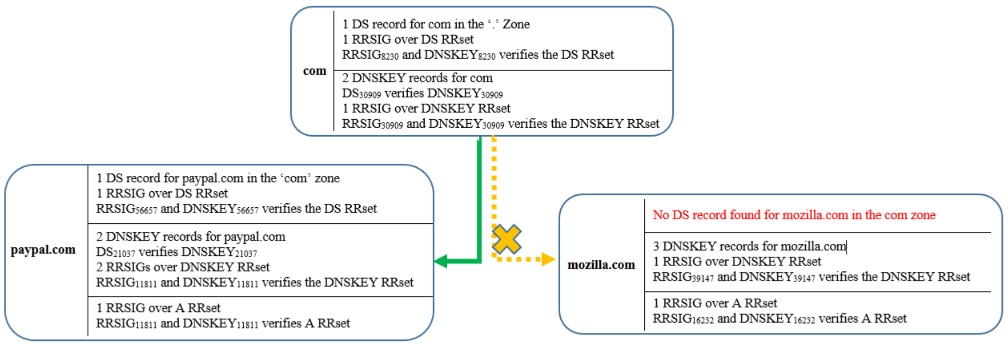

com
1 DS record for com in the '.' Zone
1 RRSIG over DS RRset
RRSIG8230 and DNSKEY8230 verifies the DS RRset
2 DNSKEY records for com
DS30909 verifies DNSKEY30909
1 RRSIG over DNSKEY RRset
RRSIG30909 and DNSKEY30909 verifies the DNSKEY RRset
paypal.com
1 DS record for paypal.com in the 'com' zone
1 RRSIG over DS RRset
RRSIG56657 and DNSKEY56657 verifies the DS RRset
2 DNSKEY records for paypal.com
DS21037 verifies DNSKEY21037
2 RRSIGs over DNSKEY RRset
RRSIG11811 and DNSKEY11811 verifies the DNSKEY RRset
1 RRSIG over A RRset
RRSIG11811 and DNSKEY11811 verifies A RRset
mozilla.com
No DS record found for mozilla.com in the com zone
3 DNSKEY records for mozilla.com
1 RRSIG over DNSKEY RRset
RRSIG39147 and DNSKEY39147 verifies the DNSKEY RRset
1 RRSIG over A RRset
RRSIG16232 and DNSKEY16232 verifies A RRset

Figure 12: paypal.com and mozilla.com

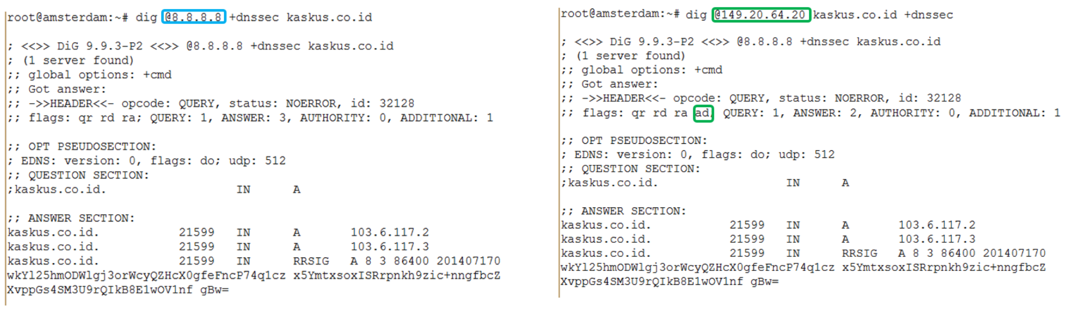
The 4-week time period that our experiments took place was not enough to monitor the DNSsec adoption rate effectively. On the one hand, the deployment status is relative low, probably because of the overhead and complexity that DNSsec introduces that make administrators not willing to configure it, regardless of the security benefits that adds to the DNS service. From the other hand, our results showed that there is a slight increase in the DNSsec adoption rate that was in accordance with other measurements, being made in different time periods.

5.3 DLV registry

There are some situations that the parent zone has not been signed or is not ready to accept DS records, here the child can use the DLV (DNSsec Look-aside Validation) registry. DLV provides an alternate chain of trust to allow any zone added to the DLV registry to be validated.

In our result 195 TLDs do not have DS records in root Zone such as, “aero”, “ir”, “rs”, “travel”. For domains with no AD flag, we repeated our experiments with another resolver to check the DLV records. Among all, only 46 domains are using DLV.

In Figure 14 you can see that the “id” and “co.id” domains do not have DNSsec-enabled while their child kaskus.co.id has DNSsec-enabled. Here, the child uses DLV records to create the chain of trust via the DLV registry as you can notice in Figure 13.



```
root@amsterdam:~# dig @8.8.8.8 +dnssec kaskus.co.id

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec kaskus.co.id
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32128
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;kaskus.co.id.                  IN      A

;; ANSWER SECTION:
kaskus.co.id.           21599   IN      A       103.6.117.2
kaskus.co.id.           21599   IN      A       103.6.117.3
kaskus.co.id.           21599   IN      RRSIG   A 8 3 86400 201407170
wkYl25hmODWlgj3orWcyQZHcX0gfeFncP74q1cz x5YmtxsoxISRrpnkh9zic+nngfbcZ
XvppGs4SM3U9rQIkB8E1wOV1nf gBw=
```

```
root@amsterdam:~# dig @149.20.64.20 kaskus.co.id +dnssec

; <<>> DiG 9.9.3-P2 <<>> @8.8.8.8 +dnssec kaskus.co.id
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32128
;; flags: qr rd ra ad, QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;kaskus.co.id.                  IN      A

;; ANSWER SECTION:
kaskus.co.id.           21599   IN      A       103.6.117.2
kaskus.co.id.           21599   IN      A       103.6.117.3
kaskus.co.id.           21599   IN      RRSIG   A 8 3 86400 201407170
wkYl25hmODWlgj3orWcyQZHcX0gfeFncP74q1cz x5YmtxsoxISRrpnkh9zic+nngfbcZ
XvppGs4SM3U9rQIkB8E1wOV1nf gBw=
```

Figure 13: kaskus.co.id

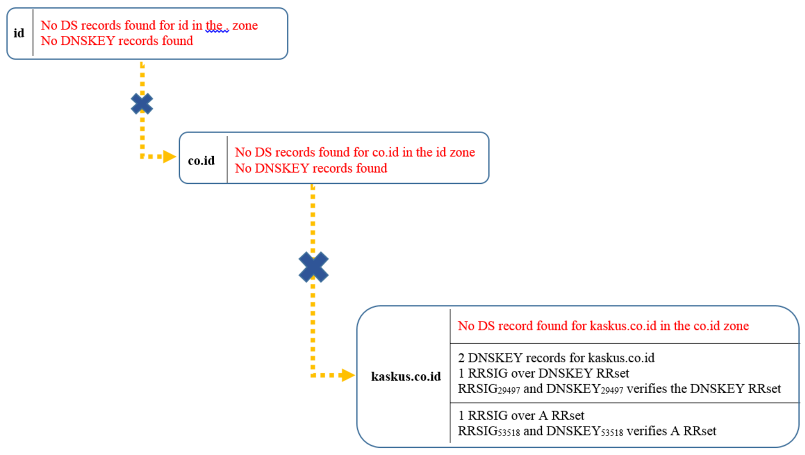

id
No DS records found for id in the . zone
No DNSKEY records found
co.id
No DS records found for co.id in the id zone
No DNSKEY records found
kaskus.co.id
No DS record found for kaskus.co.id in the co.id zone
2 DNSKEY records for kaskus.co.id
1 RRSIG over DNSKEY RRset
RRSIG29497 and DNSKEY29497 verifies the DNSKEY RRset
1 RRSIG over A RRset
RRSIG53518 and DNSKEY53518 verifies A RRset

Figure 14: kaskus.co.id

5.4 Additional experiments

In our DNSsec-enabled domain set we performed some additional experiments such as algorithm choice for DNSKEYs, use of NSEC or NSEC3 records and signature lifetime. The purpose of these experiments is to reveal, if the administrators are using best practices for configuring these parameters.

5.4.1 Algorithms

Initially, we checked the signed zones to evaluate what algorithms were mostly used. Our results is shown in Table 1 and Figure 15.

Algorithm	Number of Algorithm	Number of Domains	Percentage(%)
RSA-SHA1	5	2974	30.1
DSA-NSEC3-SHA1	6	3	0.03
RSA-NSEC3-SHA1	7	3351	33.8
RSA-SHA256	8	3554	35.6
RSA-SHA512	10	33	0.33
ECDSA Curve P-256 with SHA 256	13	1	0.01

Table 1: Algorithm used

As we can see, roughly 1/3 of the domains use the default algorithm RSA-SHA1, roughly 1/3 use the RSA-NSEC3-SHA1 and roughly 1/3 use RSA-SHA256 algorithm. All of these algorithms are considered secure. The only non-secure algorithm is RSA/MD5 which was not used by any domain.

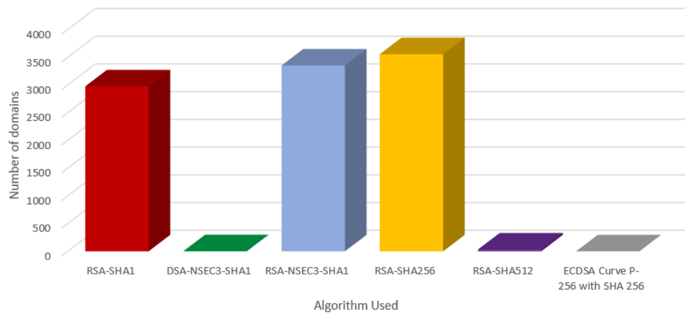


Figure 15: Algorithm

5.4.2 NSEC(3)

In our second experiment, we investigated how many domains were using NSEC and how many domains are using NSEC3 as shown in Figure 16. NSEC3 offers the same functionality as NSEC while it is more secure and prevents zone enumeration by ordering the cryptographic hashes of owner names rather than returning them in clear text. From our result 58% of signed domains have NSEC3 records while 42% of them have NSEC.

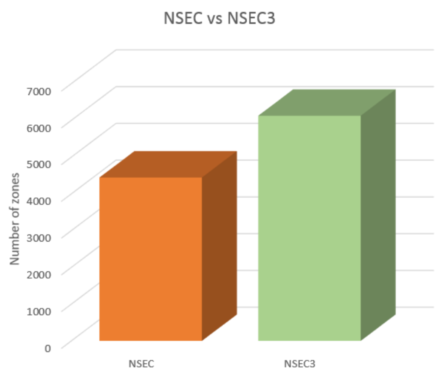


Figure 16: NSEC and NSEC3

5.4.3 Signature Lifetime

The signatures in DNSsec have a lifetime value that expire after a certain amount of time. Keys with a long effective period are particularly sensitive as they will represent a more valuable target and be subject to attack for a longer time than short-period keys. So, a reasonable effective period for KSKs can be 1 year, despite that the expected lifetime is 5 years and for ZSKs a reasonable effective period can be 30 days, despite that the expected lifetime is 90 days.

For our last experiment, we extracted the signature lifetime from our data and the results can be seen in Figure 17. This number varies between 2 and 3600. Roughly 1/3 of the domains have lifetime of 30 days, for their RRSIG timestamps.

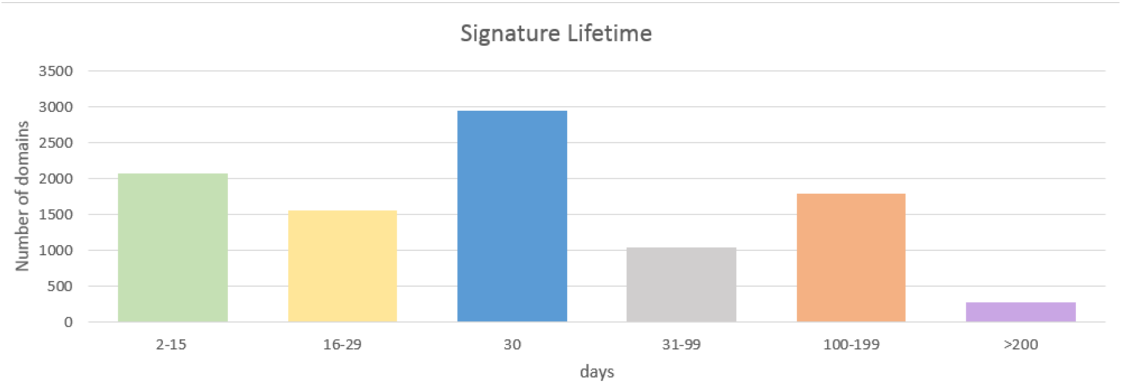


Figure 17: Signature lifetime

As DNSsec is highly depending on the precise time, the administrators must be sure about the precise time of their servers which can be configured via NTP.

5.5 DNSsec Misconfiguration

In this section we categorise the most common problems occurred in deploying and maintaining DNSsec in which we reached from our experiments, providing the errors that lead us to these problems and possible explanations behind them.

- Missing DS (Figure12)
  - None of the DNSKEY records could be validated by any of the DS records in the parent zone
  - The DNSKEY RRset was not signed by any keys in the chain-of-trust
- Mismatch DS
  - None of the DNSKEY records could be validated by any of the DS records in the parent zone
  - The DNSKEY RRset was not signed by any keys in the chain-of-trust

  In some cases, the parent includes a DS RR while the child is not publishing the corresponding DNSKEY anymore. We found 25 domains which was once DNSsec-enabled but disabled it without removing the DS record in the parent zone. Note that this situation is different from the previous one and it indicates that the child is insecure. This error may also happen when the KSKs have been expired and cannot be deemed as a SEP, or there is a bad KSK rollover.
- Missing DNSKEY
  - A DS record was found in the parent zone but its corresponding DNSKEY was not found
  - None of the RRSIG and DNSKEY records validate the DNSKEY RRset
  - DNSKEY RRset was not signed by any keys in the chain-of-trust

  If the DNSKEY referenced in the RRSIG, is not available in the DNSKEY RRset, the result becomes a bogus. This might occur if the authoritative server doesn't support DNSsec or the server is running an inconsistent version of the zone.
- Missing NSEC

  This might happen because the zone was not signed properly, so it doesn't have all the necessary records or the server doesn't support authenticated denial of existence (NSEC or NSEC3)
- Missing RRSIG

  This might be missing because the RRSIGs were erroneously deleted from the zone itself or some of the authoritative server haven't deployed DNSsec.
- Bogus RRSIG
  - DNSsec signatures did not validate the RRset

If the zone was signed with different keys than the ones that are published in the zone data we encounter the bogus RRSIG or if, for example, a DNS-based load balancer sends different A RRset for request but neglects to create the RRSIGs corresponding to RRset in the responses.

- Expired RRSIG
  - DNSsec signatures did not validate the RRset
  - The RRset was not signed by any key in the chain-of-trust

When a DNS zone is signed, the zone administrator specifies a time in the future that the signatures will expire at that time. If the administrator forgets to resign the zone, which refreshes the signatures, before the signature expiration time, the signatures are considered invalid and resolvers will not use them to validate the zone data.

In Figure 18 you can see an example of this bug.

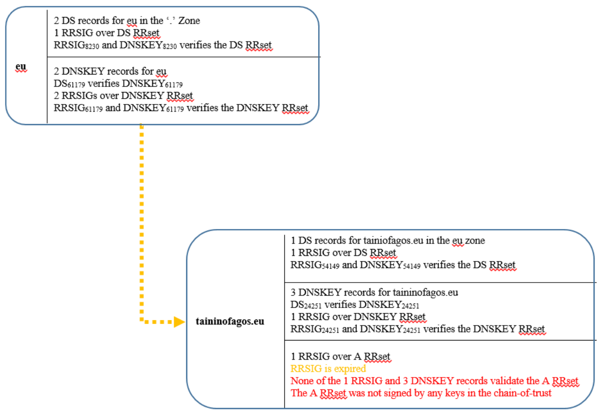
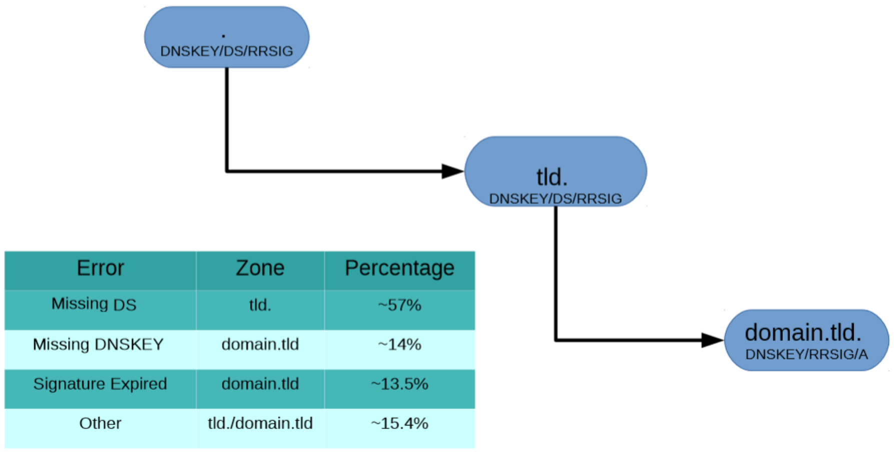

eu
2 DS records for eu in the '.' Zone
1 RRSIG over DS RRset
RRSIG$_{8230}$ and DNSKEY$_{8230}$ verifies the DS RRset
2 DNSKEY records for eu
DS$_{61179}$ verifies DNSKEY$_{61179}$
2 RRSIGs over DNSKEY RRset
RRSIG$_{61179}$ and DNSKEY$_{61179}$ verifies the DNSKEY RRset
taininofagos.eu
1 DS records for tainiofagos.eu in the eu zone
1 RRSIG over DS RRset
RRSIG$_{54149}$ and DNSKEY$_{54149}$ verifies the DS RRset
3 DNSKEY records for taininofagos.eu
DS$_{24251}$ verifies DNSKEY$_{24251}$
1 RRSIG over DNSKEY RRset
RRSIG$_{24251}$ and DNSKEY$_{24251}$ verifies the DNSKEY RRset
1 RRSIG over A RRset
RRSIG is expired
None of the 1 RRSIG and 3 DNSKEY records validate the A RRset
The A RRset was not signed by any keys in the chain-of-trust

Figure 18: domain with expired RRSIG

The results of our experiments showed (Figure 19) that most of the DNSsec errors are due to missing DS records, while the missing of DNSKEY and expired RRSIGs are the second and third highest contributors to DNSsec validation failure.



Error	Zone	Percentage
Missing DS	tld.	~57%
Missing DNSKEY	domain.tld	~14%
Signature Expired	domain.tld	~13.5%
Other	tld./domain.tld	~15.4%

Figure 19: Misconfiguration errors

Most of the invalidated, incomplete and bogus data is a result of misconfiguration done by administrators' negligence. Otherwise, it may happen via some active network attackers which tamper the data on the fly or at the zone or even in cache at recursive resolver. Storing private keys in a secure way in order to prevent unauthorized users to access them at administrator's side, and checking the existence of AD flag in the answers at client's side mitigate the success of attacks. Besides that, we have to check and monitor the trust anchors at resolvers regularly to be sure that these anchors are valid and are not expired.

6 Conclusion and Recommendation

DNS is a key element of the Internet architecture. As it is notoriously insecure, DNSsec is designed to mitigate this flaw and ensure integrity and authentication of data while increasing complexity to old DNS.

In this project we investigated the 930000 most popular website domains on the Internet. We monitored the data in their authoritative nameservers as well as the data returned by validating resolvers in order to check the healthiness of the zones and the domains. We found some inconsistencies in nameservers belonging to domains which made the returned answer non-deterministic.

Our survey showed that nearly 1% of domains are using DNSsec, and only 76% of the DNSsec-enabled domains are valid DNSsec domains. The other remaining domains are categorized as insecure and bogus ones depending on the type of their chain of trust.

As the deployment of DNSsec is recommended, a thorough understanding of the protocol accompanied by proper configuration, regular monitoring and maintenance are essential and also complex. Misconfigurations are abound that sometimes lead to eliminate the domains on the Internet while they have been requested by validating resolvers. We categorized the misconfiguration according to their place if they happened at the zone side (RRSIG missing, expired, bogus) or the parent side (DS missing and mismatch) and tried to find out those faulty domains for each misconfiguration.

For helping administrators to handle DNSsec during deployment and maintenance, there are a number of open-source DNSsec tools [4] [5] designed with different goals which can check the syntax of the zones, manage the keys and rollover, and automate the signing process of the zones. These tools are highly recommended to administrators and even users in order to avoid and mitigate the effects of misconfigurations. Figure 20 indicates where each tool can be used.

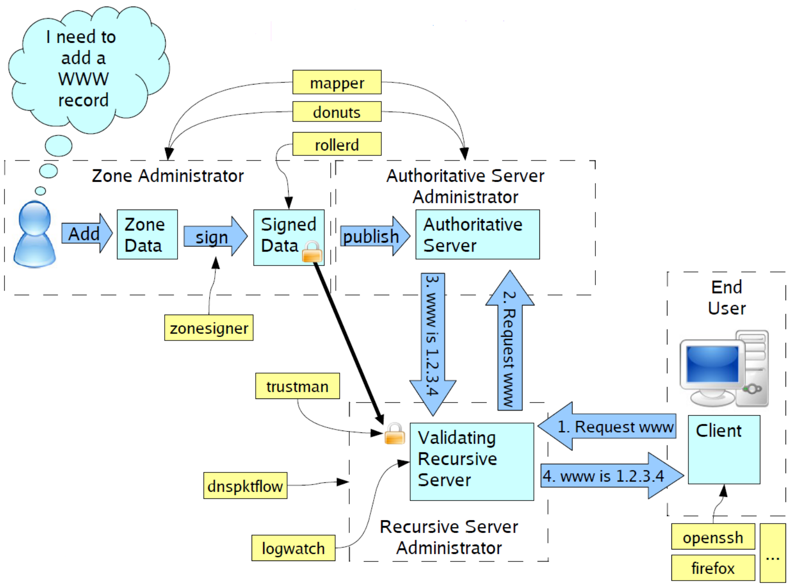


Figure 20: DNSSEC-Tools

7 Future Work

For future work, we have these ideas in mind:

- Working on the resolvers which actually validate DNSsec records to find out what is the percentage of DNSsec-aware resolvers
- Investigate in available tools designed for handling DNSsec and trying to improve them
- Investigate the effect of DANE, DNSsec doesn't provide security against on-path attackers. The attacker can intercept the traffic between the client and server. Here DANE-based certificates combined with DNSsec will provide security against those attackers.

[4]See `http://www.dnssec-tools.org`.

[5]See `https://www.nanog.org/meetings/nanog44/presentations/Sunday/Hardacker_Tools_DNSSEC_N44.pdf`.

References

[1] Cricket Liu and Paul Albitz. DNS and BIND (5th Edition). O'Reilly Media, Inc., 2006.

[2] Casey Deccio. Quantifying and Improving Dns Availability. PhD thesis, Davis, CA, USA, 2010.

[3] Kadjo Tanon Lambert, Souleymane Oumtanaga, Kone Tiemoman, Abba Brice, and Pierre Tety. Deployment of dnssec: Problems and outlines. In RCIS, pages 343–348. IEEE, 2008.

[4] Yu Sun, Ru Juan Liu, and Yi Liu. Research and implementation of dnssec monitoring system. In ICMLC, pages 2145–2150. IEEE, 2010.

[5] Eric Osterweil, Dan Massey, and Lixia Zhang. Deploying and monitoring dns security (dnssec). In Proceedings of the 2009 Annual Computer Security Applications Conference, ACSAC '09, pages 429–438, Washington, DC, USA, 2009. IEEE Computer Society.

[6] Eric Osterweil, Dan Massey, and Lixia Zhang. Observations from the dnssec deployment. In Proceedings of the 2007 3rd IEEE Workshop on Secure Network Protocols, NPSEC '07, pages 1–6, Washington, DC, USA, 2007. IEEE Computer Society.

[7] P. Mockapetris. Domain names - concepts and facilities. IETF RFC1034, November 1987.

[8] P. Mockapetris. Domain names - implementation and specification. IETF RFC1035, November 1987.

[9] M. Graff J. Damas and P. Vixie. Extension mechanisms for dns (edns(0)). IETF RFC6891, April 2013.

[10] S. Kaufman D. Eastlake 3rd. Domain name system security extensions. IETF RFC2065, January 1997.

[11] D. Eastlake. Domain name system security extensions. IETF RFC2535, March 1999.

[12] M. Larson D. Massey S. Rose R. Arends, R. Austein. Dns security introduction and requirements. IETF RFC4033, March 2005.

[13] M. Larson D. Massey S. Rose R. Arends, R. Austein. Resource records for the dns security extensions. IETF RFC4034, March 2005.

[14] M. Larson D. Massey S. Rose R. Arends, R. Austein. Protocol modifications for the dns security extensions. IETF RFC4034, March 2005.

[15] D. Eastlake 3rd P. Vixie, O. Gudmundsson and B. Wellington. Secret key transaction authentication for dns (tsig). IETF RFC2845, May 2000.

[16] D. Eastlake 3rd. Dns request and transaction signatures (sig(0)s). IETF RFC2931, September 2000.