DNSSEC Misconfigurations: How incorrectly configured security leads to unreachability

van Adrichem, NLM; Reyes Lua, A; Wang, X; Wasif, M; Fatturrahman, F; Kuipers, FA

DOI
10.1109/JISIC.2014.12

Publication date
2014

Document Version
Accepted author manuscript

Published in
Joint Intelligence and Security Informatics Conference Proceedings

DNSSEC Misconfigurations: How incorrectly configured security leads to unreachability

Niels L. M. van Adrichem, Antonio Reyes Lúa, Xin Wang, Muhammad Wasif,
Ficky Fatturrahman and Fernando A. Kuipers
Network Architectures and Services, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
{n.l.m.vanadrichem@, {a.reyeslua, x.wang-15, m.wasif, f.fatturrahman}@student., f.a.kuipers@}tudelft.nl

***Abstract*—DNSSEC offers protection against spoofing of DNS data by providing authentication of its origin, ensuring integrity and giving a way to authenticate denial of existence by using public-key cryptography. Where the relevance of securing a technology as crucial to the Internet as DNS is obvious, the DNSSEC implementation increases the complexity of the deployed DNS infrastructure, which may manifest in misconfiguration. A misconfiguration not only leads to silently losing the expected security, but might result in Internet users being unable to access the network, creating an undesired unreachability problem.**

In this paper, we measure and analyze the misconfigurations for domains in four zones (.bg, .br, .co and .se). Furthermore, we classify these misconfigurations into several categories and provide an explanation for their possible causes. Finally, we evaluate the effects of misconfigurations on the reachability of a zone's network. Our results show that, although progress has been made in the implementation of DNSSEC, over 4% of evaluated domains show misconfigurations. Of these misconfigured domains, almost 75% were unreachable from a DNSSEC aware resolver. This illustrates that although the authorities of a domain may think their DNS is secured, it is in fact not. Worse still, misconfigured domains are at risk of being unreachable from the clients who care about and implement DNSSEC verification while the publisher may remain unaware of the error and its consequences.

I. Introduction

The Domain Name System (DNS) [1] is a crucial technology for the functioning of the Internet as it enables communication using domain names that are easier to remember than numerical IP addresses. Among others, DNS maps human readable hostnames into IP addresses and provides a distributed database from which users can request these mappings. The popularity of this mapping system is explained by the use of Fully Qualified Domain Names (FQDNs) as the primary component of URLs with which all Internet users identify websites.

The importance of DNS lies in the fact that it is not only used by end users, but also by several other core network technologies for their everyday operation [2], such as Telephone number mapping (ENUM) or SIP. However, even though DNS is one of the fundamental building blocks of the Internet, its original design in 1983 focused more on its scalability and did not include security considerations. Even as early as 1990, the first flaws in the DNS were detected and the need for protecting it were discussed for the first time [3]. This led to the publishing in 1997 of the Domain Name System Extensions (DNSSEC) standard and its refinement in 2005 [4]. Moreover, after the discovery in 2008 by Dan Kaminsky of a fatal exploit in the DNS, the urgency to adopt a technology such as DNSSEC became evident [5].

DNSSEC, in a broad sense, offers protection against illegitimately falsifying data stored in the DNS by providing authentication of its origin, ensuring its integrity and giving a way to authenticate denial of existence by using public-private key cryptography. To make sure that the user receives authentic replies, DNSSEC deploys cryptographic keys. With private keys, digital signatures are generated for resources which can be verified by their public counterparts.

A. Motivation and problem definition

At the time of writing, there are 111069731 registered .com domains, and 279841 of them are signed [6]. For DNSSEC to work as intended, it must be deployed at all levels of the DNS architecture. Therefore, its adoption by all involved actors in the DNS resolution process is essential for its success. One big step was given in July 2010 when the DNS root zone was signed [7]. This enabled the resolvers to configure the root zone as a trusted anchor and allowed to validate the complete chain of trust for the first time. Nevertheless, even though 84% of existing domains could already be using DNSSEC, as more and more Top Level Domains (TLD) are being signed, just less than 1% of authoritative domain name servers have implemented it [8]. Most commonly cited causes for this problem are that:

- the implementation of DNSSEC increases the complexity for the management of the deployed DNS infrastructure, and that
- a misconfiguration might result in Internet users being unable to reach the protected network [9].

This unreachability problem may become a bigger concern when a new group of TLDs (up to 1400) [10] start rolling out the next few years, since all of them must implement DNSSEC and misconfigurations of the zone files of domains could potentially hide them from the Internet.

Despite the importance of the stated problem, there does not exist sufficient information on the current status of DNSSEC deployed zones. Therefore, in this paper, we measure the status of several DNSSEC-enabled production zones measuring both

the level of DNSSEC implementation and the correctness of DNSSEC configuration. We think that doing experimental research on DNSSEC is of great value but, in order to get a deeper insight, it should also be complemented with an analysis of the most common problems DNSSEC is experiencing in day-to-day production environments. Performing an analysis on real production data from operational zones brings a better understanding on the current status of DNSSEC deployment. Moreover, it also helps to define the biggest challenges that need to be overcome for this technology to succeed.

Initially, we focus on .bg, .br, .co, and .se domains. We have chosen those since their complete domain list is browsable as explained in subsection II-C. The .se zone draws its importance for analysis as it was the first DNSSEC signed zone [11] in 2005, 5 years prior to the root domain. Similarly .co, which represents regional host names for Colombia, is also popular for company, corporation and commerce host names. Furthermore, .br has one of the largest DNS registries in South America. Finally, we added the .bg domain to compare with a cc-TLD that has signed its zone less recently.

B. Outline

This paper is organized as follows. Section II gives a summary on the internal workings of DNS. Section III presents work related to DNSSEC misconfigurations measurements and highlights their proposed methods and shortcomings. In section IV we present the used measurement tools, while sections V and VI discuss the measurement environment and present and analyze the results from the measurements. Finally, section VII concludes this paper.

II. DNS AND DNSSEC

In this section, we summarize the functions and internal organization of the Domain Name System (DNS) and its Security Extensions (DNSSEC). In short, DNS provides a directory of typed values that are named by hierarchically structured names. It is, among others, responsible to map domain names, such as the host component of a URL, to IP addresses. Due to the wide use of domain names it performs a critical function in today's Internet. In subsection II-A we discuss how DNS implements a distributed lookup directory. Subsection II-B discusses the most relevant extensions and possible errors in configuring DNSSEC.

A. DNS

The Domain Name System (DNS), first standardized in 1983 [1], primarily offers a distributed database storing typed *values* by *name*. Each such record is called a resource record (RR), often shortly referred to as "record", and contains the following ordered properties:

- The Name of the record.
- The Type of the record, describing the type of the record stored in the record. Popular records include, the A-, AAAA-, MX-, CNAME- and SPF-records, respectively storing an IPv4 address, IPv6 address, the responsible mail exchange hostname for that domain, an alias referring to another hostname and domain specific rules regarding spam policies according to the Sender Policy Framework protocol.
- The Class code, specifying a specific name space scope. Although multiple codes exist, usage of values different from IN for Internet are uncommon.
- The Time-to-live, specifying in seconds how long intermediate or recursive DNS servers may cache that specific record, often defaulting to a zone's TTL.
- The length of the RDATA field, used for communication protocol implementation.
- The RDATA field, containing the record's actual stored data.

Most often, DNS is used to request mappings from computer hostnames to IP addresses. In order to support such a high frequency of requests, DNS employs a tree-wise hierarchy in both names and database structure. A so-called Fully Qualified Domain Name (FQDN) consists of multiple name components specifying the location of its records in a tree of databases. Clients, such as home and business PCs, connect to a local recursive (often caching) DNS server that traverses the tree to receive the requested information. In general, the traversed tree-wise structure of DNS consists of 3 layers: (1) The root-layer, a set of nameservers named [a-m].root-servers.net. (2) The Top-Level-Domain (TLD) layer. (3) The authoritative layer.

Every recursive DNS server has a list of root servers, called the "root hints file", containing a list of all root-server hostnames and IP addresses. For means of load balancing and geographic distribution of requests, anycast addresses are used to deploy multiple servers per hostname. The root-servers themselves do not contain any mappings for FQDNs, but instead refer to Top-Level-Domain (TLD) servers responsible for the requested TLD by replying with an NS-record containing the nameserver of the next layer and its IP-address in an A-record. The 2 top-most used types of TLDs are the generic TLDs such as the domains .com, .net, .edu and .org, as well as country-code TLDs whose last name suffix refers to country specific sites such as .nl for the Netherlands, .uk for Great-Britain, etc. These TLD-servers once again refer to the next layer which is generally authoritative for that domain name and returns the requested mapping. Where further recursion is possible, commonly 3 steps are sufficient. The relation between the name and the place of its records in the distributed tree is summarized in figure 1.

B. DNSSEC

Although DNS has proven to be very scalable, the architecture shows many possibilities for both un- and intended malicious behavior and attacks. It is fairly easy to tune-in and mangle with DNS requests and replies by executing a so-called man-in-the-middle attack, hence secretly redirecting the client to obscure locations or falsely denying existence of resources. This, for example, could occur at open WiFi hotspots, where providers often offer their own, potentially malicious, DNS

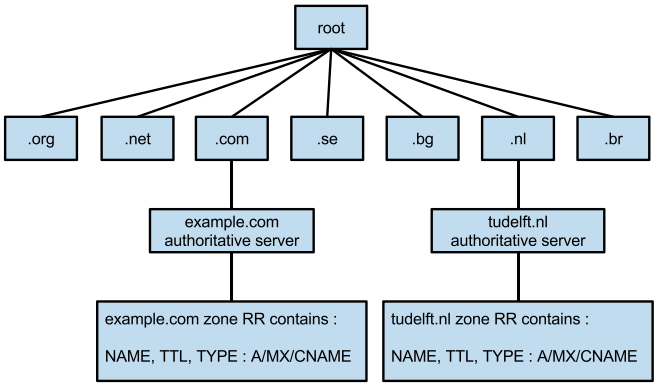

Fig. 1. A tree showing the hierarchical distributed nature of DNS names and their location in the distributed database.

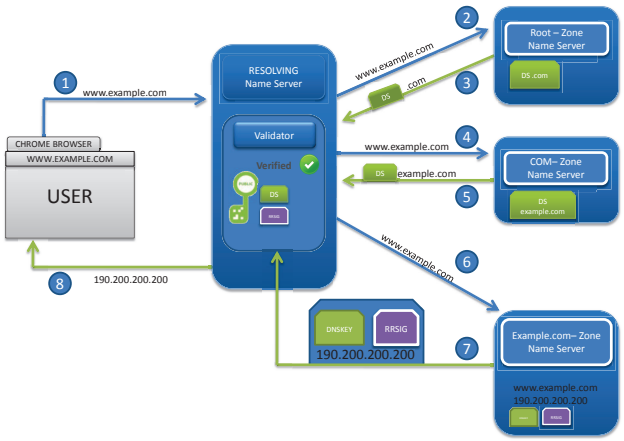

Fig. 2. Domain name resolution in DNSSEC. The domain name query www.example.com iterates from the resolver to all the name servers in the DNS tree (steps 2 to 7) until the name is resolved. Between each resolving step all the responses are verified by the validator and on success, the resolved IP address is forwarded to the user.

service. Hence, DNSSEC has been introduced to authenticate the validity of both returned RRs and non-existent records through cryptographic signing of resources.

In order to support the cryptographic signing process, each domain has multiple associated keys containing at least 1 public-private key pair. For each record set (RRset) of distinct name and type, a signature is generated using the private key and stored in an equally-named RRSIG-record, which can be verified using the domain's public key stored in a DNSKEY-record, both placed in the zone of the domain.

To confirm the authenticity of the DNSKEY, which is essential to check the authenticity of any record in a zone, its digest, called the Delegation Signer, is stored in an equally named DS-record in its parent zone. Recursively, the DS-record is signed in its local zone, and the process repeats until the root-zone is consulted. Since the root-zone has no ancestors, its DNSKEY-record is confirmed by globally publishing its digest which is referred to as the Trust Anchor [12]. As shown in figure 2, a DNS client or resolver recursively requests these records to determine authenticity of a record.

The recursive chain from Trust Anchor, to intermediate DNSKEY-, DS- and RRSIG-records authenticates each RRset of a properly configured DNSSEC domain. Part of managing public-private key pairs is refreshing them regularly to prevent malicious parties from deriving the private key. Hence public-private key pairs are equipped with an expiration date and therefore need to be renewed at regular intervals. Replacing a DNS record, however, is non-trivial due to possibly long periods of caching in clients and intermediate resolvers. Furthermore, changing one's public-key does not only involve updating one's DNSKEY-record, but also implies updating the parent DS-record whose replacement needs to be performed by the TLD, a third party, again keeping cache synchronization in mind. Therefore, key rollover can be a tedious process.

In order to ease the process, a zone may be equipped with 2 types of public-private key pairs, Key Signing Keys (KSKs) and Zone Signing Keys (ZSKs), both stored in DNSKEY-records and distinguished by a flag. KSKs concern the keys whose DNSKEY-record is confirmed by the parent's DS-record and are exclusively used to sign other DNSKEY-records in the zone, the ZSKs. ZSKs are hence used to sign all other type of resources in the zone. As ZSKs are signed by a local KSK, those public-private key pairs can rollover independent of the parent-zone. The KSK often is a longer, cryptographically more complex, key pair that does not change often. Besides a decreased need of replacing the key due to its greater complexity, KSKs only sign a limited number of resources (ZSKs) making them less prone to attacks as less in- and output of the key pair is available. The ZSK can change more often and may be cryptographically less complex if sufficiently often replaced with new keys.

C. Authentication of non-existing resources

When a DNS server is queried for a non-existent record, i.e. there is no record with requested name, it will respond with a NXDOMAIN message indicating its absence. Where DNSSEC so far authenticates existing resources, it is difficult to authenticate non-existing resources as non-existent records (1) have no corresponding signature, and (2) if for every NXDOMAIN response a signature would be computed online, key pairs would be more vulnerable to attacks. Since an NXDOMAIN message may be hijacked and replaced with a false response to silently mislead a user, it is important to authenticate non-existing resources.

In order to authenticate non-existent resources, DNSSEC introduces NSEC-records [13] containing a linked-list of existing records ordered by name, hence actively denoting non-existing namespaces. For an example domain named example.com with just 2 subnames mail. and www., the NSEC-records would look as follows:

- The first NSEC-record named example.com refers to mail.example.com indicating that mail.example.com is alphabetically the first subname of example.com, hence

actively denying the existence of any subnames that would be ordered prior to it, such as ftp.example.com.
- The second NSEC-record named mail.example.com refers to www.example.com indicating there are alphabetically no subdomain names between them, hence actively denying the existence of subsequently ordered subnames, such as news.example.com.
- The final NSEC-record named www.example.com refers back to the zone name example.com, indicating it is alphabetically the last record.

An NSEC enabled server will reply with the appropriate NSEC-record to requests for non-existing resources. Each existing NSEC-record, of whom as many exist as names exist for a domain, is ultimately signed by an RRSIG-record to confirm authenticity of the claimed non-existence. A property of the NSEC-records is that they can be iterated to gather a *complete list of valid subnames* for a domain or TLD, a process called zone-walking. We have extensively used this method to extract the lists of domain names from our selected TLDs.

However useful to our research, publishing the complete list of available resources does not relate to the occasional request of non-existent resources and may even raise concerns on privacy. Therefore, the NSEC3 additions hash the zone-specific name-component (i.e. mail. and www.) prior to ordering, and generate a recursive linked-list of NSEC3-records containing hashed values [14]. In order to authenticate non-existence of a resource, the DNS-server will hash the requested zone-specific name component and return the NSEC3-record indicating the non-existent range to which it belongs. Hence, proving non-existence without giving away existing names.

III. Related work

In this section, we discuss previous work on DNSSEC. We found most studies focus on the performance of DNSSEC, such as latency, delay, or resource load on the server, rather than on misconfigurations. In [2], the authors analyze the availability of DNSSEC resolution and service. One related study focuses on quantifying and improving DNSSEC availability [9]. The authors first identify what kind of misconfigurations in DNSSEC can affect a DNS query request. They find what are the potential failures due to DNSSEC misconfiguration, and then they create a metric to quantify those DNSSEC misconfigurations. They classify DNSSEC misconfigurations into three categories:

1) Zone (missing/expired/invalid RRSIGs covering zone data, or missing DNSKEY RRs required to verify RRSIGs).
2) Delegation (bogus delegations because of lack of appropriate DNSKEYs in the child zone corresponding to DS RRs in the parent zone, or insufficient NSEC RRs to prove an insecure delegation to a resolver).
3) Anchor (stale trust anchors in a resolver, which no longer match appropriate DNSKEYs in the corresponding zone).

They analyze only 1456 signed zones out of which 194 show to be misconfigured [15]. Out of these, most of the misconfigurations are related to zone data which corresponds to the first class of misconfigurations. However, authors do not explain why this was the case and what were the main technical causes for such misconfigurations. The class 1 misconfigurations arise due to missing or outdated RRSIGs or DNSKEYs and, as it was explained in section I, that the deployment of DNSKEY and RRSIGs for each record in the zone file is the responsibility of the zone administrator. Technically, the administrator should always ensure the correctness and validity of RRSIGs and DNSKEY deployment. Hence, the previous work does not give insight in the causes of misconfiguration, nor its effects in authenticity confirmation and reachability by a DNSSEC aware resolver. Furthermore, the analysis was done in 2010, 4 years ago when DNSSEC was in an earlier stage of deployment compared to now in 2014 when DNSSEC is more widely implemented. We believe it to be very useful to do such an analysis for a wide range of zones in order to find out if the same type of misconfigurations can be found in production zones. In this report, we analyze 22437 signed zones from .bg, .br, .co and .se domains.

IV. Measurement tools

There exist several different tools to work with DNSSEC. However, most of them are intended to be used by zone administrators in order to verify their own zone file before publishing it and require the user to have the complete zone file in order to perform their tests. Examples of such tools include Verisign's jDNSSEC [16] or NLnetLabs' LDNS [17]. We selected a set of tools that are able to perform tests over a list of several domains without possession of their zone files, that is to say, from the point of view of an external user. Furthermore, we selected these tools based on execution time and ease of automation to ease the process of checking a large amount of domains. We have used the following measurement tools to perform our measurements:

1) Dnsrecon [18]: A DNS enumeration program, written in Python, that allows to discover relevant information from the content of a zone. It performs several types of enumeration including zone transfer, reverse lookup, a Google lookup and, most importantly, zone walking using NSEC-records as discussed in section II-C. After testing the tool, we found its usage straightforward and effective since it provided us with the necessary means to easily and effectively retrieve all the authoritative name servers from a zone.

In the process of retrieving the domain lists for the domain, we enhanced the program with the following 3 functions: (1) To decrease the chance and impact of getting blocked by a DNS server due to excessive usage, we added support for multiple DNS servers. (2) We added the possibility to save the current state of iteration, so in case we were blocked by all DNS servers of a zone we were able to continue iteration from a different source IP-address. (3) Initially, Dnsrecon verified a domain's intent to deploy DNSSEC capability by checking whether the authoritative nameserver returned a DNSKEY-record for its hostname. Instead, we verified the intent to check whether a domain registered a DS-record at its parent.

2) DNSSEC-Debugger [19]: A web based tool developed by Verisign Labs that inspects the chain of trust for a particular DNSSEC enabled domain. It shows the step by step validation of a given domain and indicates any error or warning found in its DNSSEC configuration. We found Dnssec-debugger to be fast (analysis consumes approximately 3 seconds per domain) and ideal for automation as any domain can be inspected by executing a HTTP-request to http://dnssec-debugger.verisignlabs.com/⟨DOMAIN⟩.

3) Google's Public DNS Service [20]: Found at IP-addresses 8.8.8.8 and 8.8.4.4, Google offers a free and globally accessible DNSSEC enabled DNS resolution service, which can be used as an alternative to one's in-house or ISP provided DNS resolution server. In order to evaluate the effects of DNSSEC misconfiguration on the reachability of a domain, we assume a misconfigured DNSSEC domain to be unavailable when it does not pass Google's Public DNS Service.

V. Measurement Scenario

The measurement of the different domains consists of 4 different phases, followed by an additional 5th phase in which we evaluate the effects of misconfiguration in everyday use. The first phase consists of gathering a comprehensive list of domain names. To do this, we use Dnsrecon to perform zone-walking of the 4 NSEC-enabled TLDs .bg, .br, .co and .se, hence retrieving extensive lists of domain names from these domains.

The second phase consists of filtering the list of domain names by the intent of them being DNSSEC enabled. We assume a domain name to intend to be DNSSEC enabled when a DS-record for that domain is registered at its TLD-zone. Filtering is performed by iterating the list of domain names and performing DS-record lookups using the internal functions of Dnsrecon.

Having retrieved a list with a sufficient number of DNSSEC enabled domains, we verify their configuration using the DNSSEC-Debugger online tool from Verisign Labs. To do so, we iterate through the list, performing a HTTP-request to the appropriate URL and parse the response for further analysis. To verify the correctness of the DNSSEC-Debugger we took a sample from the results and compared with results from *dig*, part of the BIND DNS software suite [21], a Linux command-line tool used for querying DNS servers. Normally, it takes approximately 3 seconds to receive the verification result for one domain name. In order to overcome this time limitation and to speed up the process, we perform up to 10 lookups in parallel by employing multithreading.

In the 4th and last phase we categorize the misconfiguration in the following categories and subcategories, enabling analysis by the type of the misconfiguration:

1) Misconfigured DNSKEY-record(s)
   a) None found
   b) Key(s) not validated by DS-record at parent
   c) KSK(s) not validated by ZSK(s) (see subsection II-B)
   d) DNSKEY validated but expired
2) Misconfigured RRSIG(s)
   a) None found
   b) Signatures not validated by ZSK(s)
   c) Signatures validated but expired
   d) Signatures render corresponding RRset invalid
3) General DNS failure (of DNSSEC enabled domains)
   a) Time-out
   b) SERVFAIL
   c) REFUSED
   d) No SOA found
   e) (Optional) SOA Serial differs, indicating stale info at DNS server

Additionally, we added 2 unforeseen miscellaneous errors in a 4th category:

4) Other
   a) DS retracted: In the time between retrieving the list of DNSSEC enabled domains and performing these measurements, the domain has withdrawn from implementing DNSSEC.
   b) Server does not implement the resource record type DNSKEY and, therefore, is DNSSEC incapable.

Finally, after performing the initial measurements, we verified the effects of misconfiguration by requesting the A-records associated with misconfigured domains from Google's Public DNS Service which performs DNSSEC authentication verification.

VI. Results and evaluation

In this section we discuss and evaluate the results from the experimental measurements described in section V. Subsection VI-A shows the results from the first and second measurement phase, gathering domain names and measuring the integration of DNSSEC in the different zones. Subsection VI-B categorizes the misconfigurations of the zone .se into the nature of the misconfiguration. Finally, subsection VI-C analyzes the result of the misconfigurations on the availability of the domain.

A. DNSSEC Implementation

In this subsection we present the results of the first two phases of our measurements, (1) gathering domain names and (2) measuring the integration of DNSSEC within the list of zones. While gathering the lists of domain names by walking the NSEC-records, we were often blacklisted by the TLD nameservers as the excessive amount of performed DNS requests are classified as possible attacks on the service. As shown in table I, for most zones we were able to gather and analyze a considerable amount of domain names. The .br zone, however, appeared to have additional counter-measures against zonewalking. Regularly, the .br TLD nameservers would reply with an NSEC-record indicating the requested domain was the last domain name of the zone, hence terminating the zone walking process as it appeared to be finished.

Table I shows both historical statistics found on per-zone DNSSEC implementation, as well as the number of domain

TABLE I
HISTORICAL STATISTICS AND MEASUREMENT ON DNSSEC IMPLEMENTATION PER CCTLD.

ccTLD	Statistics from:	Total	DNSSEC	%	Retrieved	DNSSEC	%	Misconfigurations	%
.bg	08/2008 [22]	N/A	80	N/A	38806	162	0.42%	26	16.04%
.br	01/2014 [23]	3310972	487471	14.72%	2481	504	20.31%	2	0.00%
.co	10/2013 [24]	1560000	196	0.01%	151707	23	0.02%	6	26.09%
.se	09/2013 [25]	1292596	327684	25.35%	89772	21748	24.23%	876	4.03%
Total					282766	22437	7.93%	910	4.06%

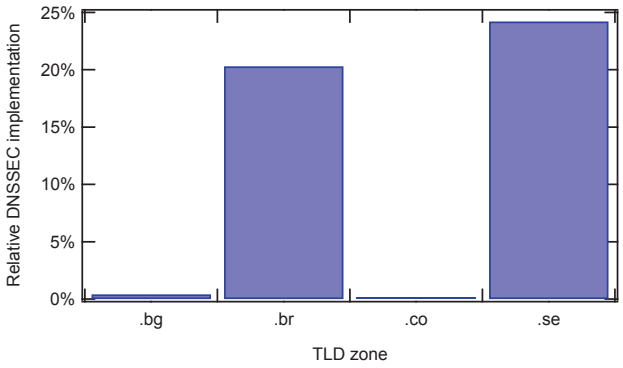

Fig. 3. Relative implementation of DNSSEC per country-code TLD.

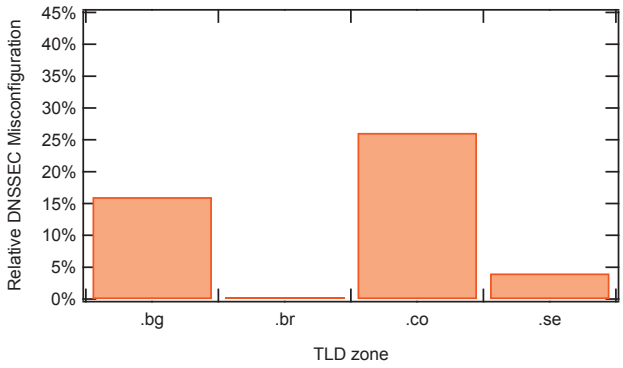

Fig. 4. Relative misconfigurations in found DNSSEC domains.

names we were able to gather and check for DNSSEC implementation. Although our lists are incomplete, we were able to confirm the relative implementation of DNSSEC for the selected zones. We found that the zones .bg and .co both have a very low implementation of DNSSEC, resulting in a very small set of DNSSEC enabled domains. For the zone .br, we found a significant amount of DNSSEC enabled domains. Due to the aforementioned zone-walking counter-measures, however, we were unable to gather a large set of DNSSEC enabled domains for the .br domain. For the zone .se, however, we were able to gather an extensive amount of domains and DNSSEC-enabled domains. Hence, we continue to further analyze the configuration mistakes found in the zone .se. Figure 3 shows the relative percentage of DNSSEC implementation per zone.

B. DNSSEC Misconfigurations

As seen in table I and figure 4, the ccTLD .se has a significant amount of found misconfigurations. Table II shows the misconfigurations related to the categories and subcategories listed in section V. As also prospected in figure 5, approximately two-thirds of the misconfigurations is related to configuration of the DNSKEY records. Slightly less than one-third of the misconfigurations is caused by missing DNSKEY records, indicating there once was an intention or maybe even a running configuration to deploy DNSSEC. However the DNSSEC configuration has never been properly configured or removed from the authoritative nameserver. Slightly more than a third of the misconfigurations indicates a Key Signing Key that is not properly signed by its Zone Signing Key as described in subsection II-B, hence breaking the chain of trust. The situation where a ZSK invalidates the zone's KSK indicates a problem with the key-rollover of the KSK, most probably it has not been resigned after it was renewed. Once DNSKEY configuration is properly done, we find little evidence in the internal configuration of the authoritative nameserver, from the category of possible RRSIG misconfigurations we only found 1 occurrence of an expired signature.

Stunningly, a third of the misconfigurations seem to revolve around general DNS misconfigurations or errors that could also occur in non-DNSSEC environments. Especially the number of reported server failures and time-outs are surprisingly high. We where unable to confirm whether these errors are strictly related to non-DNSSEC configuration, and thus unrelated to DNSSEC, or are caused by a server malfunction due to incompatibility with the DNSSEC-extended query. We did however find two occurrences with a more specific error, where the server indicated incompatibility with the DNSKEY resource record type, showing that DNSSEC-incompatibility with DNS servers once intended to perform DNSSEC is a problem. Finally, we found 14 occurrences in which the authoritative administration retracted its intention to implement DNSSEC before we were able to scan its zone for misconfiguration. In figure 5, we show the distribution among the most significant configuration errors.

C. Effects on availability

After performing the measurements, we proceeded to verify the reachability of the misconfigured domains using a DNSSEC validating resolver. For that purpose, we used Google's Public DNS Service which has implemented DNSSEC validation by default since May 2013 [20].

The result of this experiment shows that 73.86% of the

TABLE II
MEASURED MISCONFIGURATION STATISTICS FOR THE CCTLD .SE AND RESPECTIVE REACHABILITY BY GOOGLE'S PUBLIC, DNSSEC AWARE, DNS RESOLUTION SERVICE.

#	Error Category	Subcategory	Misconfigurations	%	Unreachable	%
1	**DNSKEY**	**Total**	**564**	64.38%	**477**	84.40%
a		Not found	261	29.79%	259	99.23%
b		Invalidated by DS	88	10.05%	3	3.41%
c		KSK invalidated by ZSK	215	24.54%	214	99.53%
d		Valid but expired	0	0%	N.A.	
2	**RRSIG**	**Total**	**1**	0.11%	**1**	100.00%
a		Not found	0	0%	N.A.	
b		Invalidated by ZSK(s)	0	0%	N.A.	
c		Valid but expired	1	0.11%	1	100.00%
d		Invalidate RRset	0	0%	N.A.	
3	**General DNS failure**	**Total**	**295**	33.68%	**163**	55.25%
a		REFUSED	12	1.37%	11	91.67%
b		SERVFAIL	191	21.80%	142	74.35%
c		Time-out	64	7.31%	9	14.06%
d		No SOA	4	0.46%	0	0.00%
e		SOA Serial differs	24	2.74%	1	4.17%
4	**Miscellaneous**	**Total**	**16**	1.83%	**7**	43.75%
a		DS retracted	14	1.60%	5	35.71%
b		DNSKEY RR Failure	2	0.23%	2	100.00%
#	**All categories**	**Total**	**876**	100%	**647**	73.86%

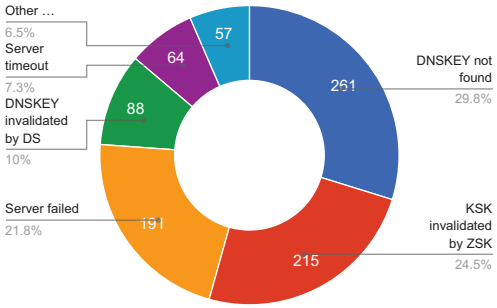

Fig. 5. Distribution of most significant DNSSEC misconfigurations.

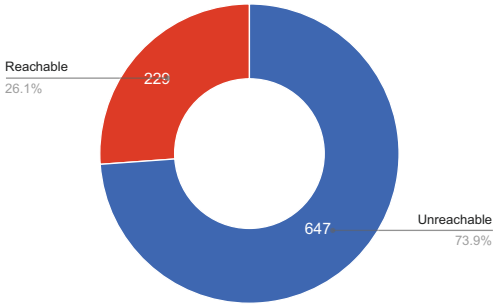

Fig. 6. Availability of misconfigured DNSSEC enabled domains.

misconfigured domains in the ccTLD .se were completely unreachable from a DNSSEC aware resolver. As also shown in figure 6, the remaining 26.14% of domains still had some misconfiguration, but those were not as severe to provoke the domain to become unavailable. To learn the impact of a misconfiguration, we correlated the (un-)reachability of each domain to its misconfiguration category in table II. Summarized in figure 7 after combining the categories with less than

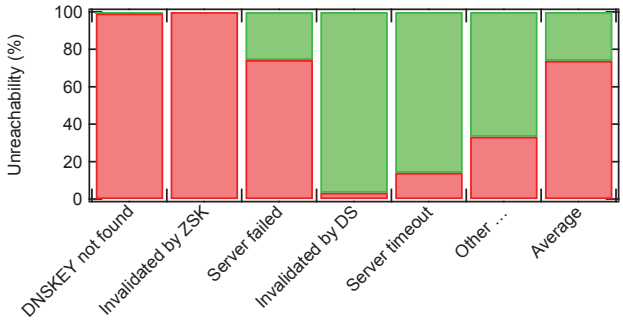

Fig. 7. Relative misconfigurations per category in the ccTLD .se.

50 misconfigurations, the impact on reachability of the most common misconfiguration types becomes clear. Concerning the DNSSEC specific misconfigurations, the impact of a missing DNSKEY record or a ZSK being invalidated by its KSK is large, nearing a 100% of unreachability. A DNSKEY invalidated by the parent DS-record indicating a potential security breach of the complete domain, however, only fails integrity checks at 3.41% of the sampled domains, even though this error may be considered as serious as the previous DNSKEY related errors. A general DNS server failure when the DNSKEY is requested leads to an unreachability level of 74.35%, similar to the overall average. Finally, we notice server timeouts are handled correctly in most cases by the caching functionality of the resolver.

VII. CONCLUSION

Having analyzed the DNSSEC misconfigurations of four zones (.bg, .br., co. and .se), our measurements show that

implementing DNSSEC is not trivial and that misconfigurations exist in large numbers. From the 282766 gathered domain names, only 7.93% show the intent to implement DNSSEC. Furthermore, over 4% of DNSSEC enabled domains show a form of misconfiguration, emphasizing the configuration complexity. Where one might expect expiration of keys to be a significant means of misconfiguration, categorization of the errors found in the .se domains shows its impact to be neglectable. Instead, most DNSSEC related misconfigurations are caused by an inconsistency concerning the DNSKEY, the main public key of a domain. In more than 99% of the cases of a missing DNSKEY or an error in the two-stage ZSK and KSK DNSKEY signing process, the error led to an unreachable domain and thus unreachable website or other network service. User availability shows to vary per type of misconfiguration. On average, 73.86% of the misconfigured domains appeared unreachable from a DNSSEC aware resolver. Hence, organizations implementing DNSSEC need to frequently verify the correct configuration of DNSSEC parameters and perhaps implement mechanisms to guarantee continuous correctness of configuration and authentic availability of their resources.

ACKNOWLEDGMENTS

The authors thank Christian Doerr and Norbert Blenn for their valuable feedback and participation in insightful discussions. Furthermore, we thank Verisign for providing us with detailed information on the verification tests performed by the Verisign DNSSEC Debugger.

This research has been partly supported by the EU FP7 Network of Excellence in Internet Science EINS (project no. 288021).

REFERENCES

[1] P. Mockapetris, "Domain names - implementation and specification," RFC 1035 (INTERNET STANDARD), Internet Engineering Task Force, Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604. [Online]. Available: http://www.ietf.org/rfc/rfc1035.txt

[2] D. Migault, C. Girard, and M. Laurent, "A performance view on dnssec migration," in *Network and Service Management (CNSM), 2010 International Conference on*. IEEE, 2010, pp. 469–474.

[3] S. M. Bellovin, "Using the domain name system for system break-ins," in *Proceedings of 5th USENIX UNIX Security Symposium, USENIX Association, Berkeley, CA*, 1995.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC 4033 (Proposed Standard), Internet Engineering Task Force, Mar. 2005, updated by RFCs 6014, 6840. [Online]. Available: http://www.ietf.org/rfc/rfc4033.txt

[5] D. Kaminsky, "Black ops 2008: It's the end of the cache as we know it," *Black Hat USA*, 2008.

[6] (2014, jan) Statdns - dns and domain name statistics. [Online]. Available: http://www.statdns.com

[7] (2014, may) Root dnssec. [Online]. Available: http://www.root-dnssec.org/

[8] (2012, dec) Dnssec securing the internet: Benefits to companies and consumers. [Online]. Available: http://www.icann.org/en/news/in-focus/dnssec/dnssec-card-03dec12-en.pdf

[9] C. Deccio, J. Sedayao, K. Kant, and P. Mohapatra, "Quantifying and improving dnssec availability," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*. IEEE, 2011, pp. 1–7.

[10] (2014, Jan.) Applicant guidebook — icann new gtlds. [Online]. Available: http://newgtlds.icann.org/en/applicants/agb

[11] (2010, Oct.) Dnssec - the path to a secure domain. [Online]. Available: https://www.iis.se/english/domains/tech/dnssec/

[12] (2014, may) Root zone dnssec trust anchors. [Online]. Available: http://data.iana.org/root-anchors/

[13] J. Schlyter, "DNS Security (DNSSEC) NextSECure (NSEC) RDATA Format," RFC 3845 (Proposed Standard), Internet Engineering Task Force, Aug. 2004, obsoleted by RFCs 4033, 4034, 4035. [Online]. Available: http://www.ietf.org/rfc/rfc3845.txt

[14] B. Laurie, G. Sisson, R. Arends, and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence," RFC 5155 (Proposed Standard), Internet Engineering Task Force, Mar. 2008, updated by RFCs 6840, 6944. [Online]. Available: http://www.ietf.org/rfc/rfc5155.txt

[15] C. Deccio, "Quantifying the impact of dnssec misconfiguration." in *DNS-OARC Workshop (2)*, Oct. 2010. [Online]. Available: https://www.dns-oarc.net/files/workshop-201010/Casey-2010-10-14-dns-oarc-orig.pdf

[16] (2014, may) Verisign jdnssec-tools. [Online]. Available: http://www.verisignlabs.com/jdnssec-tools/

[17] (2014, may) ldns. [Online]. Available: http://www.nlnetlabs.nl/projects/ldns/

[18] (2014, Jan.) Dnsrecon. [Online]. Available: https://github.com/darkoperator/dnsrecon

[19] (2014, Jan.) Dnssec analyzer. [Online]. Available: http://Dnssec-debugger.verisignlabs.com

[20] (2014, May) Public dns - google developers. [Online]. Available: https://developers.google.com/speed/public-dns/

[21] C. Liu and P. Albitz, *DNS and Bind*. " O'Reilly Media, Inc.", 2006.

[22] D. Kalchev. (2014, Jan.) Dnssec implementation in .bg. [Online]. Available: http://www.cctld.ru/files/pdf/Presentations_09-09/17-45_dnssec%20Bulgaria.pdf

[23] (2014, Jan.) Home - dominios - estatisticas. [Online]. Available: http://registro.br/estatisticas.html

[24] G. Romero, "Dnssec deployment in .co," in *ICANN48 DNSSEC Workshop*, Nov 2013.

[25] (2014, Jan.) Growth .se — .se. [Online]. Available: https://www.iis.se/english/domains/domain-statistics/growth/