The State of DNSSEC Provisioning Automation

Wilco van Beijnum
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
a.c.w.vanbeijnum@student.utwente.nl

ABSTRACT

Research into DNSSEC has shown that the adoption of DNSSEC has been quite slow so far. To make DNSSEC easier to deploy, RFC 7344 and RFC 8078 were released which specify provisioning automation for DNSSEC. However, there is a lack of research into the current state of provisioning automation. This paper will look into the support for provisioning automation in three areas. First of all, software support is analyzed. Here, the research shows that scanning for provisioning automation records is still limited, but there are already some open-source implementations available. For operators of second-level domain names, most authoritative DNS software feature some form of support for publishing provisioning automation records. Second, support for provisioning automation at the parent side will be analyzed, where the focus will be put on TLD registry support. Here, the research shows that parent side support is still very limited. However, multiple registries and a registrar have signalled support for provisioning automation in the future. Third, daily snapshots of DNS zones were analyzed to investigate provisioning automation support at the child side. Here, the research shows that very few domains currently use provisioning automation. Additionally, some misconfigurations are brought to light. Finally, a general conclusion is drawn on the current state of DNSSEC provisioning automation.

Keywords

DNS, DNSSEC, DNS Security Extensions, provisioning automation, CDS, CDNSKEY, TLD registry

1. INTRODUCTION

The Domain Name System (DNS) plays a critical role in the way that the Internet is used today. It translates domain names, like *example.com*, into IP addresses [1]. Unfortunately, the DNS protocol has security vulnerabilities [2]. To improve the security and safeguard the integrity of the DNS, DNS Security Extensions (DNSSEC) were created. These extensions are defined in RFC 4033 [3]. DNSSEC improves the security of DNS by adding signatures to all responses. However, DNSSEC is quite complex to set up and manage. An important factor in this is the involvement of the registrar when using DNSSEC. This is one of the reasons why DNSSEC has seen relatively slow adoption in some areas [4].

In order to make DNSSEC easier to set up and manage, two more specifications have been released that define DNSSEC provisioning automation, namely RFC 7344 [5] and RFC 8078 [6]. These specifications allow domain name operators to enable and disable DNSSEC, without the involvement of the registrar of the domain name. Additionally, key rollovers (changing the signing keys) to, for example, maintain a secure zone or to upgrade to another signing algorithm, are easier to perform. This research will look into the current state of adoption of these two RFC specifications in three different areas, namely DNS server software, the parent side and the child side. Here, the parent side scans the DNS zone of the child side for provisioning records. This can be done by the Top Level Domain (TLD) registry or the registrar. After this, the DNS zone of the registry is updated accordingly. For the parent side, the main focus will be on TLD registries. For the child side, the focus will be on support at domain names.

The results of this research are expected to be useful for registries and registrars that want to implement provisioning automation support. Open-source software is discussed that can be utilized. Additionally, some possible misconfigurations to pay attention to when implementing provisioning automation are brought to light.
Additionally, the research is expected to be useful for DNS providers and domain administrators that want to utilize provisioning automation. The authoritative DNS server software packages that support provisioning automation are given, as well as details on the degree of support. Additionally, the TLDs that currently support provisioning automation are listed, and implementation details for these TLDs are given.

First, I will provide background information on DNS, DNSSEC and DNSSEC provisioning automation. After this, I discuss related work. Then, I will describe the methodology for investigating the current state of provisioning automation at DNS server software, the parent side and the child side. After this, I will display the results of the research. Finally, I will draw conclusions on the current state of DNSSEC provisioning automation.

2. BACKGROUND

In order to understand DNSSEC provisioning automation and the relevance of this research, some background knowledge is required. This section will give some background information about the functioning of the Domain Name System, DNSSEC and provisioning automation.

DNS

The Domain Name System (DNS) translates domain names

35th Twente Student Conference on IT July 2nd, 2021, Enschede, The Netherlands.

into IP addresses [1]. The DNS is structured into so-called zones. Using this structure, every parent zone can point to the authoritative DNS server of a child zone. At the end of this chain, the authoritative DNS server of the domain can return the IP address of the domain name. In this way, a domain name can be resolved by recursively asking DNS servers what the IP address of a domain name is, or the IP address of a server that might know the answer.

Unfortunately, the DNS system is vulnerable to cache poisoning [2]. By using a man-in-the-middle attack, a malicious user can inject fake DNS records into the cache of a recursive DNS resolver. This vulnerability has been mitigated by randomizing transaction IDs and source port numbers, which makes these attacks significantly harder to execute, however, not impossible [7].

DNSSEC
To improve the security and safeguard the integrity of the DNS, DNSSEC was created [3]. DNSSEC improves the security of DNS by adding signatures to all responses. The public signing keys used to sign the responses are stored in the `DNSKEY` record of the responding DNS server. To validate these keys and make sure they are not compromised, a so-called chain of trust is used. A hash of the `DNSKEY` record is stored in the parent DNS zone inside the Delegation Signer (`DS`) record, which is signed by the `DNSKEY` record of that parent zone, which is again stored as a hash in that zone's parent, all the way up to the root DNS zone. The signing key of the root DNS zone is validated using a so-called trust anchor. This is done by for example adding the key to the validating recursive resolver or obtaining it via some other trusted path. A visualization of this chain of trust can be found in Figure 1. Here, the diagram shows that the root zone signs the *.nl* zone, which signs the *utwente.nl* zone. It is noteworthy that there are two signing keys at each zone, a Key Signing Key (KSK) that signs the `DNSKEY` records, and a Zone Signing Key (ZSK) that signs the other records.

RFC 4033 does not specify an automated way to communicate the signing keys to the parent zone in order to store the `DS` record. In practice, this communication takes place via the registrar of the domain name using out-of-band methods. However, many registrars do not support DNSSEC yet. This is an important reason DNSSEC has not been deployed on a large scale yet, as studies in 2017 by *T. Chung et al.* have shown [4]. Additionally, performing a key rollover is quite complex since the `DS` record of the TLD registry has to be updated via the registrar. This problem is especially prevalent when the registrar is not also managing the DNS server of the domain. In this case, a new signing key that is generated by the DNS provider needs to be passed to the registry via the registrar, such that the `DS` record can be updated. Some registrars have implemented a form to do this [8], however, for other registrars this is still a manual process [4].

DNSSEC provisioning automation
To improve this situation, RFC 7344 has been released in 2014, which solves the problem of the registrar having to forward the DNSKEY to the registry [5]. It does this by defining two DNS record types that can be added to the child's DNS zone, namely `CDS` (Child Delegation Signer) and `CDNSKEY` (Child DNS Key). The values of these records correspond with a new `DS` or `DNSKEY` record that the domain should use. These records are then read by the parent, which can modify its `DS` record accordingly. The parent is free to use either one of these records to either manually calculate the new `DS` value from the `CDNSKEY` or use the precalculated value in `CDS`. The reading of these records can take place in two ways. First of all, automated polling can be used, where the parent side will periodically request the `CDS` and `CDNSKEY` records of all its children. Another possibility described in section 6.1 of RFC 7344 is to use a polling trigger, such as a web form, after which the child records will be read. Because the `CDS` and `CDNSKEY` records are signed with the old `DNSKEY`, the parent can be sure that these records are authentic.

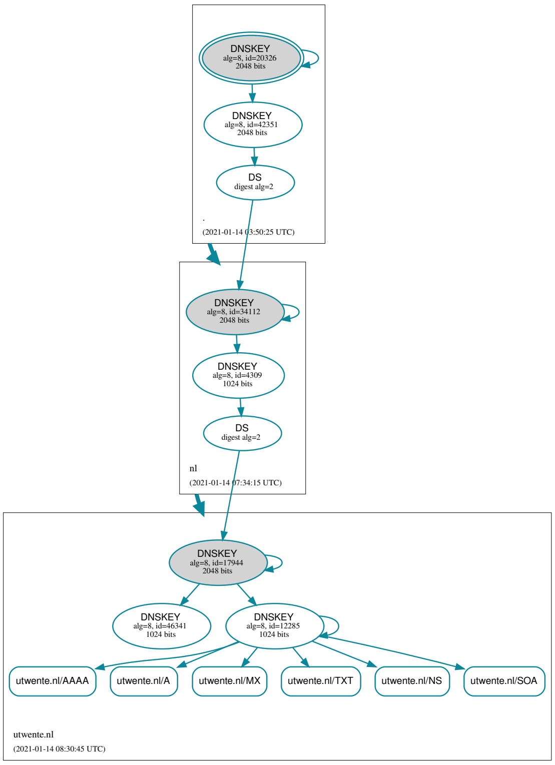

Figure 1. A visualization of the DNSSEC chain of trust of the domain *utwente.nl* [9].

While RFC 7344 has improved the process of key rollovers by no longer requiring the involvement of the registrar, there are still some limitations of this standard. First of all, when the domain is not using DNSSEC yet, RFC 7344 features no way of communicating the `DNSKEY` to the registry to enable DNSSEC, since the `CDS` and `CDNSKEY` records cannot be signed with a trusted `DNSKEY`. Additionally, it features no way of disabling DNSSEC. These limitations were finally addressed in RFC 8078 [6]. It describes ways to enable DNSSEC by using the `CDS` and `CDNSKEY` records. Several possible checks are listed to validate the authenticity of the provisioning automation records before adding a `DS` record. For example, extra checks can be done, such as email confirmation (*Accept with Extra Checks*), or the parent can wait for a certain amount of time during which the records cannot change (*Accept after Delay*). Additionally, RFC 8078 defines a way to disable DNSSEC by providing `0` as the DNS Security Algorithm Number, which may for example be useful when migrating to a different DNS provider.

3. RELATED WORK

Currently, no academic research is available on provisioning automation. To the best of our knowledge, the only scientific documents on provisioning automation are the aforementioned RFC specifications [5, 6]. Additionally, a draft was found on how to use provisioning automation in combination with a multi-signer group, where multiple entities sign the same zone [10]. Apart from these documents on provisioning automation, several research papers on the state of DNSSEC in general could be found [4, 11, 12, 13, 14, 15]. These papers show the slow adoption of DNSSEC before RFC 8078 [6] and RFC 7344 [5] were more widely adopted. Additionally, the complexity of setting up and managing DNSSEC is made clear. Finally, some insight can be gained from these papers on how the state of DNSSEC was analyzed. Some of these papers made use of the OpenINTEL dataset [4, 11, 12], which is a measurement platform that creates daily DNS record snapshots of a large number of domains [16]. This dataset will also be used in this research. Some of these papers do mention provisioning automation, but none of them have researched the current state.

In addition to the aforementioned papers on the state of DNSSEC, a paper on the key rollover process without using provisioning automation was found [17], which also shortly mentions provisioning automation but does not go into detail on its current state. Apart from scientific sources, some relevant presentation slides related to provisioning automation have also been found [18, 19, 20]. These slides give some insight into the current state of DNSSEC as well as provisioning automation. However, these slides do not give a complete picture of the state of provisioning automation.

4. METHODOLOGY

In this section, the methodology will be discussed for the research into three areas that are relevant to provisioning automation. First of all, the methodology for researching software support will be discussed. Afterwards, the methodology for research into support at the parent side will be discussed. Finally, the methodology for investigating support at the child side will be discussed.

4.1 Provisioning automation in DNS server software

To analyze the support for provisioning automation in different authoritative DNS server software packages, a list of software was first compiled. This was done by first compiling a list of available authoritative DNS server software based on a list on Wikipedia [21]. Additionally, Google was used to find TLD registry management software. After this, a selection was made based on which software is still in active development. This was determined by looking at the date of the last release of the software or, in the case of open-source software, the last commit of the source code repository. The software packages with a last update or commit of at most 6 months ago were selected.

After compiling a list of software packages to analyze, the documentation and, in the case of open-source software, the source code repositories were searched for the occurrence of one of the following terms: "CDS", "CDNSKEY", "7344", "8078". The first two search terms refer to the DNS resource records related to provisioning automation. The last two search terms refer to the RFC documents in which provisioning automation is defined [5, 6]. Based on the results of these searches and analysis of the source code of the open-source software packages, a list of currently supported features was composed per software package. After this, it was decided to further explore the support of provisioning automation in the software packages *BIND 9* and *KnotDNS*. These two software packages were chosen because of their most extensive support for provisioning automation. They were installed in a Docker container on a server and the functionality for provisioning automation was tested.

In addition to authoritative DNS server software, some registry management software was also discovered and added to the list. Additionally, GitHub was searched for open-source `CDS` and `CDNSKEY` scanners.

4.2 Provisioning automation at the parent side

Since parent side support for provisioning automation is, to our knowledge, currently only implemented at TLD registries, the focus was placed on analyzing support at TLD registries. To this end, a list of TLDs to analyze was first compiled. For this, a list of the top 100 TLDs by number of domains was retrieved from zonefiles.io, which actively crawls the Internet for domain names [22]. This list can be found in Appendix A. After compiling this list, the IANA Root Zone Database [23] was used to find the registries that correspond to these TLDs. After this, literary research was performed to find if any of these registries support provisioning automation. This was done by searching Google with the search terms "site:<website of the registry>" and ".<tld>" in combination with the search terms "CDS", "CDNSKEY", "7344" and "8078". The found results were then analyzed for any mentions of current or upcoming support for provisioning automation.

In addition to looking for information about support for provisioning automation on the Internet, a questionnaire was also sent out to the registries. Using the contact information on the IANA website and on the websites of the registries, a list of email addresses was compiled of the registries of the top 100 TLDs. In addition to this, the questionnaire was shared with the CENTR [24] and DNS-OARC [25] groups. CENTR is a council of European national TLD registries. DNS-OARC is an analysis and research center that focuses on DNS, of which many TLD registries are a member. This questionnaire asked registries if they currently support provisioning automation, or if they plan to do so in the future. It also investigated what the main limitations are in implementing provisioning automation. For TLDs that support provisioning automation, some technical questions were also asked, such as how often domains are scanned for changes in the `CDS` and `CDNSKEY` records [26].

After investigating the support and planned support for provisioning automation, the registries that already support provisioning automation were further investigated. To this end, a *BIND 9* DNS server was set up and domain names for these TLDs were registered with this DNS server as the nameserver. After this, DNSSEC was enabled using the `CDS` and `CDNSKEY` records. With DNSSEC enabled, several `CDS` and `CDNSKEY` records were tested to validate whether these registries properly process these records.

During the testing as described above, several key and algorithm rollovers were performed. Support for the following algorithms was tested: `RSASHA1-NSEC3-SHA1`, `RSASHA256` and `ECDSAP256SHA256`. Additionally, publishing multiple `CDS` and `CDNSKEY` records at once and publishing a Zone Signing Key were tested.

4.3 Provisioning automation at the child side

To analyze the support of provisioning automation at domains and DNS providers, datasets from the OpenINTEL project were used [16]. The OpenINTEL project creates daily snapshots of the DNS zones of a large number of domains for research purposes. Two types of data were analyzed. Firstly, a dataset was obtained for a TLD that already supports provisioning automation, namely *.ch*. This dataset was provided under NDA. Secondly, a dataset was obtained that contains snapshots of the DNS zones of the Alexa top 1 million domains. This data is publicly available for download [27].

There are some additional details worth mentioning about the analysis of the data. To identify the DNS provider of a website, the nameserver (`NS`) record of the domain was used. Using the domain name of the nameserver of a domain, the used DNS software could also be identified. This was done by using the *dnspython* library [28] to get the `CHAOS TXT` record `version.bind` from the nameserver. For many DNS software packages, this returns the name and/or version of the used DNS package [29, 30, 31, 32]. However, the returned version can be overwritten in the configuration files of DNS software. In this case, the used software cannot be identified.
To analyze whether a domain used provisioning automation to enable DNSSEC, the time between adding a `CDS` record to the DNS zone and the publication of a DS record in the parent DNS zone was calculated. For the *.ch* TLD, DNSSEC is enabled after the `CDS` record is published for three days, but this could also be measured as two or four days depending on the time the OpenINTEL data was gathered. Because of this, it is assumed that DNSSEC was enabled using provisioning automation if the aforementioned time period is between two and four days. Of course, DNSSEC could still have been manually enabled in this time period. However, we assume that this is not the case for the majority of the domains that match these criteria.

5. RESULTS

In this section, the results will be discussed. First, the support for provisioning automation in parent and child side DNS server software will be discussed. After this, support for provisioning automation at the parent side will be discussed. Finally, support at the child side will be discussed.

5.1 Provisioning automation in DNS server software

In this section, the results of the analysis on the state of provisioning automation support in DNS server software will be discussed. For this analysis, a list of software has first been compiled, as discussed in Section 4.1. This list can be found in Table 1. This table gives a list of the software that was analyzed, as well as a basic list of the features they support. In the rest of this section, these features will be discussed in more detail. This will be done by differentiating between parent side and child side features. On the parent side, tools to automatically update `DS` records based on the `CDS` and `CDNSKEY` records of child DNS zones will be discussed. On the child side, support for the `CDS` and `CDNSKEY` records will be discussed.

Parent side
On the parent side, there are no new record types added by RFC 7344 or RFC 8078. All of the analyzed software packages support DNSSEC and thus the `DS` record. Because of this, support for provisioning automation can be added using external scripts or tools that update the `DS` records in accordance with the `CDS` or `CDNSKEY` records of the child zones.

To update the DS records of the parent zone, the `CDS` and `CDNSKEY` records first need to be read from the child zone. This can happen at a trigger or a certain interval [5]. After this, the `DS` records need to be updated. *BIND 9* is the only authoritative DNS software package that was found to have a tool that helps with the updating of `DS` records, namely `dnssec-cds` [53]. To this tool, a file containing the relevant child DNS records can be passed. These records will then be validated and the `DS` record for the child will be updated according to the `CDS` records in the child's zone. A similar tool is requested for *PowerDNS* via an issue [54]. However, this issue has been opened on 22 September 2015 and has not seen any significant updates since then.

Software	Reviewed version	Parent side Auto update DS	Child side Auto update CDS/CDNSKEY
Authoritative DNS software			
BIND 9 [33]	9.13.3	Using tool	Yes
Knot DNS [34]	3.0.5	No	Yes
MaraDNS [35]	3.5.0019	No	No
NSD [36]	4.3.6	No	No
PowerDNS Authoritative Server [37]	4.4.1	No	Using tool
YADIFA [38]	2.4.2	No	No
BIG-IP DNS [39]	16.0.1	No	Using UI
Registry software			
FRED [40]	2.42	Yes	Not applicable
Nomulus [41]	20210520-RC00	No	Not applicable
Scanner software			
cdnskey-scanner [42]	14-12-2020	Not applicable	Not applicable
rcdss [43]	0.7	Not applicable	Not applicable

Table 1. Overview of the tested DNS server software

09-2014	RFC 7344 released [5]
02-10-2015	`CDS` and `CDNSKEY` generation added to PowerDNS [44][45]
05-11-2015	`CDS` and `CDNSKEY` generation added to BIND 9 [46]
03-2017	RFC 8078 released [6]
29-05-2017	Automatic `CDS` and `CDNSKEY` generation, and key rollover added to Knot DNS [47] [48]
05-10-2017	*dnssec-cds* tool added to BIND 9 [49]
10-11-2017	AKM and *cdnskey-scanner* added to FRED [50]
06-11-2019	*dnssec-policy* configuration option added to BIND 9 [51]
07-06-2021	*rcdss* released [52]

Table 2. Timeline of software development related to provisioning automation

In addition to authoritative DNS software, two registry management software packages were also analyzed, namely *FRED* by CZ.NIC [40] and *Nomulus* by Google [41]. Among many other features, these management software packages provide zone file generation. Because of this, parent side provisioning automation support can also be implemented in these software packages. For *FRED*, the zone generation feature also includes automatic keyset management (AKM), which can automatically update the `DS` records of DNS zones based on the `CDNSKEY` records of child DNS zones [55]. It is noteworthy that *FRED* does not support the `CDS` record. The automatic updating is done by first scanning all the DNS zones of the registered domains using the *cdnskey-scanner* tool. After getting the `CDNSKEY` record of a domain it updates the zone file of that domain in the registry accordingly. The scanning is performed at a configurable interval. By default, the maximum time between two scans of a domain is two days, and the acceptance time to enable DNSSEC is seven days [56]. More information about the scanning functionality of FRED can also be found in the paper by *M. Shchavleva* [57]. *Nomulus* does support DNSSEC but does not support provisioning automation.

The previously mentioned *cdnskey-scanner* tool [42] can also be used as a standalone command-line tool. Using this tool, a list of domains can automatically be scanned for changes to the CDNSKEY records. To do this at a set interval, an external tool like *cron* [58] has to be used. After providing a list of domain names and a list of the current DS records to this tool, this tool will scan the DNS zones of these domains and output the changes to the CDNSKEY records in a predefined format. This output can then be used by another tool or script to update the DS records. A similar tool has been released by RIPE NCC for CDS records, the *RIPE NCC CDS scanner* (*rcdss*) [43].

Child side

The support for provisioning automation on the child side first of all depends on the new CDS and CDNSKEY records being supported [5]. Luckily almost all DNS software packages support these records. Only *MaraDNS* was found to not support these records yet, even though it is still under active development.

The software packages *NSD* and *YADIFA* only provide minimal support CDS and CDNSKEY. They can be manually added to the DNS zone of a domain but there are no tools or automated functionalities built into these packages to create or update these records. Interestingly, no issues requesting more extensive support could be found on Github for these software packages.

BIND 9, *Knot DNS*, *PowerDNS* and *BIG-IP DNS* all have a tool or configuration option to automatically generate CDS and CDNSKEY records. Of the open-source software, *PowerDNS* was fastest in implementing this, adding the `pdnsutil set-publish-cds` and `set-publish-cdnskey` commands in October 2015 [44, 45]. These commands enable automatic generation and publication of respectively CDS and CDNSKEY records for a provided zone based on the DNSKEY of that zone. A similar feature was added one month later to *BIND 9*, which added the generation of the CDS and CDNSKEY records to commands used to sign the DNS zone, such as the `dnssec-signzone` command [49]. Finally, *BIG-IP DNS* features a command and a web UI option to add CDS and CDNSKEY records to the DNS zone [59, 60].

In May 2017, the first version of automated CDS and CDNSKEY record generation was added to *Knot DNS* [47, 48]. These records are automatically generated by default when signing is enabled. Since then, support for provisioning automation has been further extended. Out of the tested software packages *Knot DNS* currently features the most extensive support for provisioning automation. As of writing this, there are two signing modes available in *Knot DNS*, manual and automated key management [61].

In manual key management mode, the zone manager has to manually generate and manage keys, for example using the `keymgr` tool included with *Knot DNS*, after which *Knot DNS* will automatically generate the appropriate DNS records. In automated key management mode, the management of keys is completely automated and managed by *Knot DNS*. A lifetime for the KSK and ZSK can be set, as well as a preferred algorithm and key sizes to use for key generation. After this *Knot DNS* will automatically generate keys and sign the zone using these keys. In addition to this, after the lifetime of a key has expired, *Knot DNS* will automatically perform a key rollover. It does this by generating new keys, updating the relevant DNS records of the zone, and then periodically polling the parent DNS zone to see if the DS record is updated. Once the DS record contains the new key, the old keys are removed from the child DNS zone.

A similar feature was added to *BIND* in version 9.16. This version added the `dnssec-policy` configuration option [62]. Using this configuration option with a registry that offers provisioning automation support allows for automatic enabling of DNSSEC and automatic key rollovers. However, unlike *Knot DNS*, *BIND* does not check for updates to the DS record in the parent zone at the moment of writing. So, to complete a rollover, the `rndc dnssec -checkds published` command needs to be executed once the DS record is updated. Like *Knot DNS*, *BIND 9* also still supports the original method of manually signing zones.

5.2 Provisioning automation at the parent side

In this section, provisioning automation on the parent side will be discussed. The results of the literary research will first be displayed, after which the results of the questionnaire will be explained. Finally, some additional information about provisioning automation for specific registries will be given.

Literary research

To analyze the current support for provisioning automation at TLD registries, literary research was performed, and a questionnaire was sent out. First of all, a list of 100 TLDs was compiled, as described in Section 4.2. It was found that these 100 TLDs are managed by 73 different registries. For these 73 registries, literary research was performed on their current support for provisioning automation, and future plans to support it. Based on this research, the following TLD registries were found to support provisioning automation: *.ch/.li* [63], *.cz* [64] and *.sk* [65]. Additionally, several sources have been found indicating that *.cr* supports provisioning automation [18, 20, 19], although no official sources from NIC CR could be found. It is noteworthy, however, that *.cr* does use the *FRED* software [40]. In addition to the registries already supporting provisioning automation, two registries have also indicated that they will add support in the future, namely *.se/.nu* [66] and *.pt* [67].

Questionnaire results

After completing the literary research of these 100 TLDs, a questionnaire was set up and sent to the registries of these TLDs, as well as the CENTR [24] and DNS-OARC [25] groups. This questionnaire enquired registries for details about their current support or plans for future support of provisioning automation. Ultimately, this questionnaire was filled out by representatives of 20 TLD registries. The questionnaire included an option to submit results anonymously, so some results below do not mention specific registries. A general overview of the future plans of registries can be found in Figure 2.

Out of these 20 registries, two registries, *.bg* and an anonymous registry, indicated that they will not support provisioning automation. For *.bg*, it was indicated that they will not support provisioning automation because they have an alternate system in place for automatically enabling DNSSEC. Additionally, it was argued that people in an organisation that manage the DNS zone should not be authorized to change DNS records stored in the parent zone. The other registry indicated that they will not be implementing support because of opposition from registrars.

In total, 16 registries indicated that they may implement provisioning automation in the future. These registries were enquired on the main limitations when deploying support for provisioning automation. Here, lack of priority and resources seem to be the most predominant limitations. Additionally, the lack of demand for DNSSEC was

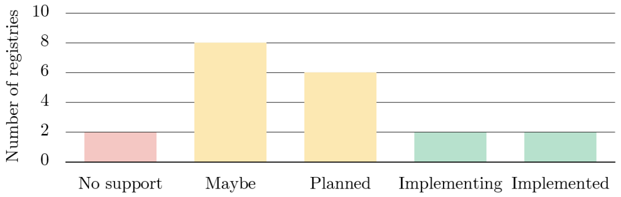

Figure 2. Questionnaire results of future plans of registries to support provisioning automation.

TLD	CDS	CDNSKEY	Bootstrap from insecure
.ch	Yes	No	3 days TCP-only
.cr	No	Yes	7 days TCP-only
.cz	No	Yes	7 days TCP-only
.li	Yes	No	3 days TCP-only
.sk	Yes	No	3 days

Table 3. Implementation details for TLD registries that support provisioning automation [20]

mentioned multiple times. Out of these 16 registries, two registries indicated that they are already working on support, namely *.cl* and *.se/.nu*. Additionally, six registries indicated that they will work on support in the future, namely *.ca*, *.fi*, *.dk* and three anonymous registries. The remaining eight registries indicated that they may support provisioning automation in the future but do not have any plans yet.

Finally, the two registries that indicated they already support provisioning automation, *.ch/.li* and *.cz*, were enquired on the details of implementing provisioning automation. The main reason for adding provisioning automation support that was indicated is to make DNSSEC easier to configure and deploy. Some challenges that *.cz* encountered while implementing support are communication with registrants, implementing scanning such that it can be executed from multiple locations in a parallel manner and dealing with registry locks. When a registry is locked, information can only be updated with additional authentication. At *.cz*, *FRED* is used to scan for changed `CDNSKEY` records [50], which is also mentioned in Section 5.1. Meanwhile, *.ch/.li* uses an in-house solution inspired by the *cdnskey-scanner* software from CZ.NIC [42] to scan for `CDS` records.

Implementation details

The registries that support provisioning automation were also asked how often they scan for `CDS` or `CDNSKEY` records, and which record type they use. A presentation by *O. Caletka* was found to have a clear overview of this information as well [20], which also includes the other registries that support provisioning automation, but that did not fill out the questionnaire. This overview can be found in Table 3. It is worth noting that all registries that support provisioning automation also allow the enabling and disabling of DNSSEC as defined in RFC 8078 [6]. Interestingly, not all registries use the same record type. Some registries use the `CDS` record, while others use the `CDNSKEY` record. For a registry, the `CDS` record is easier to process, since it can just be copied and saved as a `DS` record. The advantage of using the `CDNSKEY` is that the registry can choose what digest to use. There are currently no registries that are known to support both records. All registries use the *Accept after Delay* method for enabling DNSSEC. For some registries, this delay is 3 days, while other registries have a delay of 7 days.

In addition to the questions about provisioning automation, the questionnaire also inquired which DNS software is used at the registry. The most used software package is *BIND*. 18 registries indicated using this. Following that are the software packages *NSD* with 11 registries and *Knot DNS* with 7 registries. It is noteworthy that many registries use more than one software package. Because of this, it might make more sense to add support for provisioning automation scanning to a registry package like *FRED* or to develop a standalone application.

In addition to the previously mentioned information, some other details were found that may be useful for DNS providers or domain name administrators. SWITCH, the registry of *.ch* and *.li* has published a document with details on their provisioning automation implementation [63]. It is noteworthy that only a subsection of the available algorithms and digests are supported for usage in provisioning automation. In addition to this document, a webpage is also available that can be used to check the provisioning automation status of a domain name [68]. CZ.NIC, the registry of *.cz* has implemented provisioning automation in their open-source registry management software *FRED* and points to the documentation of this software package for details on the implementation [55]. It has also been disclosed that CZ.NIC is very liberal on the acceptance of `CDNSKEY` records and does not check if updating the `DS` records breaks the chain of trust. Because of this, they do not completely adhere to the RFC specifications [5, 6]. This approach is being reconsidered, however. For the *.sk* and *.cr* TLDs, no further implementation details could be found.

Testing

For the testing of provisioning automation at several TLDs that currently support it, three domain names were registered. This way, the registries of *.ch/.li*, *.cz* and *.sk* could be tested for their implementation. Since *.cr* also used *FRED* it was chosen not to test this registry.

Due to some issues that were encountered and the relatively long time it takes to process the changes by the registries (1-7 days), not many different configurations were able to be tested. First of all, some problems were encountered at *.cz*. When publishing multiple `CDNSKEY` records, they would sometimes not be accepted. This was traced back by CZ.NIC to a mismatch in the ordering of the `CDNSKEY` records and entries in their database. This issue has now been fixed [69]. Additionally, some problems were encountered at *.ch* and *.sk*. Here, updated provisioning automation records would only be accepted after several days instead of one day. During the other days, the *CDS Status Check* website of *.ch* [68] would show the error "The CDS RRSet was not signed by a valid key-signing key (KSK)". At SWITCH, this problem was ultimately traced back to the long Time To Live (TTL) that we used for the `CDS` records and the fact that BIND 9 caches records up to 7 days by default, causing the provisioning automation processor to use possibly outdated, cached values. SWITCH has meanwhile reduced the maximum caching time on their resolver infrastructure and is looking at providing a clearer error message in the CDS Status Check tool in this case. The problems at *.sk* were likely also caused by this problem, although we were unable to get in contact with someone at SK-NIC to confirm this.

For both the *.ch* and *.sk* domain, `DS` records for a Zone Signing Key (ZSK) were not rejected. For *.cz*, this was unable to be tested due to the aforementioned problem. However, CZ.NIC indicated that, aside from this problem, their `CDNSKEY` processor should have accepted the key. No problems were encountered with the acceptance of the tested algorithms as listed in Section 4.2.

Other parent side implementations
In addition to the aforementioned TLDs, RIPE NCC has recently also added support for provisioning automation [70, 71, 43]. RIPE NCC manages Reverse DNS servers [72] as well as ENUM servers [73], which now support `CDS` records to perform key and algorithm rollovers. Currently, RIPE NCC does not support enabling DNS from an insecure zone using provisioning automation as described in RFC 8078 [6]. The deleting of a record with algorithm `0` is supported, however.

Provisioning automation does not have to be implemented at a registry level. It can also be implemented at a registrar level, which can communicate the relevant records to the registries using the currently in place methods, such as by using Extensible Provisioning Protocol (EPP) [74]. To the best of our knowledge, this is not used in practice yet. However, GoDaddy has disclosed they are planning to implement provisioning automation [75]. In this way, DNS providers will be able to utilize provisioning automation for TLDs that do not support it yet.

5.3 Provisioning automation at the child side

To analyze the current usage of provisioning automation, the OpenINTEL dataset has been used [16]. To be able to compare the usage between domains with TLDs that do and do not support provisioning automation yet, a dataset of *.ch* domains and a dataset of the Alexa top 1 million domains (*Alexa 1M*) [27] were used.

The *.ch* dataset contains around 2.1 million unique domain names. One dataset contains snapshots from the first day of each month, from January 2021 until June 2021. The other dataset contains snapshots of every day of the month of May 2021. Unfortunately, OpenINTEL only started collecting data of all the *.ch* domains in 2021, so no older historical data is available. The *Alexa 1M* dataset contains around 800,000 unique domain names. Here, one dataset contains snapshots from the first day of each month, again from January 2021 until June 2021. The other dataset contains snapshots of 1 June of the years 2019 until 2021.

Domains using provisioning automation
For the *.ch* dataset, it was found that around 7.2% of domains use DNSSEC. For only around 0.75% of domains, a `CDS` record was found at some point in time within a month. No significant increase in provisioning automation usage in the time period of five months was found.
For the *Alexa 1M* the fraction of domains using DNSSEC is much smaller, namely 2.1%. For provisioning automation, one would expect the percentage of domains to be lower than that for *.ch* domains since *.ch* already support provisioning automation, and most other TLDs do not. However, the publication of provisioning automation records is higher, with around 2% of domains publishing a `CDS` record, and around 1.7% of domains publishing a `CDNSKEY` record at some point in time within a month. Around half of these domains publishing a provisioning automation record currently do not use DNSSEC.

Enabling and disabling DNSSEC using provisioning automation
Provisioning automation is actively used to enable and disable DNSSEC for *.ch* domains. However, it is only used for a relatively small number of domains. Around 11% of new domains start using DNSSEC within a month. However, only 0.2% of new domains likely used provisioning automation to enable DNSSEC.
Disabling DNSSEC using provisioning automation is used even less often. In the dataset, 250 domains were found with a `CDS` record with the `DELETE` algorithm. However, only around 5 domains per month were found where DNSSEC was enabled at first and disabled after the publication of a `CDS DELETE` record. For the other domains, DNSSEC was already disabled for a longer time, so these domains are likely not actively using provisioning automation.

Key usage
The dataset was also analyzed for algorithm and digest usage. This information might be useful for deciding what algorithms to support as a party that want to implement parent side provisioning automation. For both the *.ch* and *Alexa 1M* datasets, `ECDSAP256SHA256` was found to be the most used algorithm. At *.ch* domains, this algorithm was used at 95% of the domains with a `CDS` record. For the *Alexa 1M* dataset, this was a bit less, namely 59% of `CDS` and 72.6% of `CDNSKEY` records. This is partly explained by the fact that there are a lot more `DELETE` records present for the *Alexa 1M* dataset, namely 19% of `CDS` records and 24.2% of `CDNSKEY` records. The second most used algorithm is `RSASHA256`. A comparison was also made between domains with provisioning automation and domains that use DNSSEC in general. It was also found that, in the latter case, the `ECDSAP256SHA256` algorithm is used less. Instead, the algorithms `RSASHA256`, `RSASHA1` and `RSASHA1-NSEC3-SHA1` are more predominant, so domains using provisioning automation seem to be using more up-to-date algorithms.

Nameserver support
By grouping the domains in the dataset by nameserver domain and analyzing how many domains in the group use a `CDS` or `CDNSKEY`, some conclusions can be drawn on the support of provisioning automation by a DNS operator. If almost all domains publish a `CDS` or `CDNSKEY` record, the DNS provider of the nameserver is likely to automatically publish these records. If there are no domains that publish a `CDS` or `CDNSKEY` record, the DNS provider might not support provisioning automation.

Domains that almost always publish a provisioning automation record were found to be mostly based in countries with registries that already support provisioning automation, such as Switzerland or the Czech Republic. Most of the top 20 largest DNS providers in the dataset by number of domains did not seem to support provisioning automation, with a 0% provisioning automation usage. Two exceptions to this are Cloudflare and Google Cloud DNS. These providers had a provisioning automation usage of at most 24% of the domains using Google Cloud DNS and at most 8% of domains using Cloudflare, depending on the dataset. Cloudflare already announced support for provisioning automation in September 2018 [76]. Interestingly, no information could be found about provisioning automation support on Google Cloud DNS.

Misconfigured CDS and CDNSKEY records
Something to take into account as a registry when implementing provisioning automation is that `CDS` and `CDNSKEY` records are not always configured properly. In the dataset, three domains in the *.ch* dataset and one domain in the *Alexa 1M* dataset were encountered that published a `CDS` or `CDNSKEY` record for Zone Signing Keys (ZSKs) in addition to the Key Signing Keys (KSKs). These records were still published as `DS` records, however. Additionally, around 30 domains were found that published a provision-

ing automation record without a corresponding DNSKEY record. In one case, the corresponding DNSKEY was missing completely. But in all the other cases, the mismatch was caused by the CDS record having a key tag of 0. After looking up the nameservers for these domains, it was noticed that all of these domains were using the DNSimple nameserver. DNSimple was contacted in regards to this issue. At the moment of writing, they are looking into the issue. A registry may thus want to recalculate the key tag of the CDS records, which *.ch* seems to do, or ignore these records.

Nameserver software

During the data analysis, the usage of nameserver software for nameservers with domains with provisioning automation was also compared to nameservers with no domains with provisioning automation. It is expected that, in the first case, the usage of DNS software with more extensive provisioning automation support is more prevalent, as opposed to the second case. However, due to the large number of nameservers for which the used software could not be determined, this is hard to conclude. For the *.ch* dataset, 67% of domains with a CDS record uses *PowerDNS* and 31% of software is unknown. For domains without a CDS record, only 10% of domains were found to be using *PowerDNS*, and 9% *BIND 9*. However, for 80.7% of the domains, the used software could not be identified.

6. CONCLUSION

Support for provisioning automation in software packages at the parent side is still quite minimal. The software package *FRED* offers the most extensive support. However, there are also quite a few other open-source software packages available. These can be used together with, or as the basis for a custom program. On the child side, provisioning automation support in authoritative DNS software is more extensive. Several software packages offer good support for generating provisioning automation records, or even fully automate the process.

Currently, not many registries support provisioning automation. However, several registries have plans to implement it. A lack of demand, priority and resources seem to be the main limitations for most registries. Most registries do have a positive stance towards the specification, however. Additionally, several registries have indicated that they will implement provisioning automation in the future. A lack of demand could partly be solved by more actively promoting the existence of provisioning automation as a registry. For the registries that support provisioning automation, it was found that some use the CDS record, while others use the CDNSKEY record. Because of this, DNS software and DNS providers should support publishing both record types. Testing of provisioning automation showed that there are still some bugs in the processors of the tested TLDs. These problems will hopefully be solved in the future. During the testing process, the *CDS Status Check* website of *.ch* [68] was very useful, so this is something to consider implementing as a provisioning automation parent.

There are currently not many domains that publish CDS or CDNSKEY records. However, as we have seen earlier, there is already software support for provisioning automation in most popular DNS software packages. Thus, the lack of provisioning automation usage may largely be caused by either a lack of knowledge about the existence of provisioning automation or a lack of support at DNS providers. By automatically publishing CDS and CDNSKEY records as a DNS provider, the usage of provisioning automation can easily be increased. Registries and registrars implementing provisioning automation have to take into account that CDS and CDNSKEY records are sometimes misconfigured. They will need to decide on how liberal they want to be in accepting or ignoring these records.

In conclusion, provisioning automation has not seen a large scale adoption as of yet. However, as support will continue to grow at TLD registries and registrars, I expect more DNS providers will also start using provisioning automation. To gain more insight into the plans of DNS providers and registrars, future studies could look into DNS providers and registrars more extensively, by performing literary research on larger DNS providers and registrars and sending out questionnaires. Some other shortcomings of this research are the fact that only 20 TLD registries answered the questionnaire, which may not be a representative group. Additionally, more extensive manual tests could be performed to validate that TLDs have properly implemented provisioning automation.

The preliminary results of this paper were also presented at the ICANN71 conference [77].

ACKNOWLEDGMENTS

I would like to thank my supervisor, Moritz Müller, and the reviewers that reviewed this paper. Additionally, I would like to thank all the representatives of registries that filled out the questionnaire on provisioning automation support at TLDs and everyone else that has provided information for this paper. I would like to specifically thank the representatives of SWITCH and CZ.NIC for their time and help during the testing of provisioning automation on *.ch* and *.cz* domains. Finally, I would like to thank Steve Crocker from ICANN for his support of this research. This research was made possible by OpenINTEL (`https://www.openintel.nl/`), a joint project of SURFnet, the University of Twente, and SIDN.

REFERENCES

[1] P. Mockapetris. Domain names - concepts and facilities. RFC 1034, RFC Editor, Nov. 1987.

[2] S. Son. The hitchhiker's guide to DNS cache poisoning. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, volume 50, pages 466–483. Springer, Berlin, Heidelberg, 2010.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, RFC Editor, Mar. 2005.

[4] T. Chung, R. Van Rijswijk-Deij, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. Understanding the Role of Registrars in DNSSEC Deployment. *IMC '17: Proceedings of the 2017 Internet Measurement Conference*, 17, Nov. 2017.

[5] W. Kumari, O. Gudmundsson, and G. Barwood. Automating DNSSEC Delegation Trust Maintenance. RFC 7344, RFC Editor, Sept. 2014.

[6] O. Gudmundsson and P. Wouters. Managing DS Records from the Parent via CDS/CDNSKEY. RFC 8078, RFC Editor, Mar. 2017.

[7] C. R. Dougherty. VU#800113 - Multiple DNS implementations vulnerable to cache poisoning. `https://www.kb.cert.org/vuls/id/800113`, Apr. 2014.

[8] GoDaddy. Add a DS record | Domains - GoDaddy Help. `https:`

//godaddy.com/help/add-a-ds-record-23865, May 2021.

[9] DNSViz. DNSViz | A DNS visualization tool. https://dnsviz.net/, Apr. 2021.

[10] U. Wisser and S. Huque. DNSSEC automation. Internet-Draft draft-wisser-dnssec-automation-01, Internet Engineering Task Force, Feb. 2021. Work in Progress.

[11] T. Chung, R. Van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. *A Longitudinal, End-to-End View of the DNSSEC Ecosystem*. USENIX, Nov. 2017.

[12] M. Müller, W. Toorop, T. Chung, J. Jansen, and R. Van Rijswijk-Deij. The Reality of Algorithm Agility: Studying the DNSSEC Algorithm Life-Cycle. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, pages 295–308. Association for Computing Machinery, Oct. 2020.

[13] E. Osterweil, D. Massey, M. Ryan, and L. Zhang. Quantifying the operational status of the DNSSEC deployment. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, pages 231–241, New York, New York, USA, 2008. ACM Press.

[14] E. Osterweil, D. Massey, and L. Zhang. Deploying and monitoring DNS security (DNSSEC). In *Proceedings - Annual Computer Security Applications Conference, ACSAC*, pages 429–438, 2009.

[15] C. Deccio. Maintenance, mishaps and mending in deployments of the domain name system security extensions (DNSSEC). *International Journal of Critical Infrastructure Protection*, 5(2):98–103, July 2012.

[16] OpenINTEL. OpenINTEL: Active DNS Measurement Project. https://openintel.nl/, Apr. 2021.

[17] M. Müller, T. Chung, A. Mislove, and R. van Rijswijk-Deij. Rolling with confidence: Managing the complexity of DNSSEC operations. *IEEE Transactions on Network and Service Management*, 16(3):1199–1211, May 2019.

[18] O. Caletka. Implementing RFC 7344. https://xn--ondej-kcb.caletka.cz/dl/slidy/20210224-DNSWG-CDS_scanning.pdf, Feb. 2021.

[19] P. Špaček. DNSSEC security without maintenance ... with the right software and registry. https://archive.fosdem.org/2019/schedule/event/dns_dnssec_security_without_maintenance/attachments/slides/3141/export/events/attachments/dns_dnssec_security_without_maintenance/slides/3141/DNSSEC_20190203_PS_FOSDEM_automated_key_management.pdf, Feb. 2019.

[20] O. Caletka. Deployment of CDS. https://ripe82.ripe.net/wp-content/uploads/presentations/62-Deployment_of_CDS.pdf, May 2021.

[21] Comparison of DNS server software - Wikipedia. https://en.wikipedia.org/wiki/Comparison_of_DNS_server_software, Apr. 2021.

[22] Zonefiles. All active domain lists updated daily. https://zonefiles.io/, May 2021.

[23] IANA. IANA - Root Zone Database. https://www.iana.org/domains/root/db, May 2021.

[24] CENTR. CENTR. https://www.centr.org/, May 2021.

[25] DNS-OARC. Home | DNS-OARC. https://www.dns-oarc.net/, May 2021.

[26] W. van Beijnum. DNSSEC Provisioning Automation at TLD Registries. https://forms.gle/mBZsFLpJEgXmUdLaA, June 2021.

[27] OpenINTEL. Index of /data/alexa1m. https://data.openintel.nl/data/alexa1m/, June 2021.

[28] Dnspython Contributors. dnspython | dnspython. https://www.dnspython.org/, June 2021.

[29] Internet Systems Consortium. 4. BIND 9 Configuration Reference — BIND 9 documentation. https://bind9.readthedocs.io/en/latest/reference.html#built-in-server-information-zones, June 2021.

[30] PowerDNS. Authoritative Server Settings — PowerDNS Authoritative Server documentation. https://doc.powerdns.com/authoritative/settings.html#version-string, June 2021.

[31] NLnet Labs. NLnet Labs Documentation - NSD - nsd.conf.5. https://www.nlnetlabs.nl/documentation/nsd/nsd.conf/#version, June 2021.

[32] CZ.NIC. Configuration Reference — Knot DNS 3.0.7 documentation. https://www.knot-dns.cz/docs/3.0/html/reference.html#version, June 2021.

[33] Internet Systems Consortium. BIND 9 - ISC. https://www.isc.org/bind/, May 2021.

[34] CZ.NIC. Knot DNS. https://www.knot-dns.cz/, May 2021.

[35] S. Trenholme. MaraDNS - a small open-source DNS server. https://maradns.samiam.org/index.html, May 2021.

[36] NLnet Labs. NLnet Labs - NSD - About. https://www.nlnetlabs.nl/projects/nsd/about/, May 2021.

[37] PowerDNS. Welcome to PowerDNS. https://www.powerdns.com/auth.html, May 2021.

[38] EURid. HOME | YADIFA. https://www.yadifa.eu/, May 2021.

[39] F5. Deliver app performance and availability with F5 DNS | F5. https://www.f5.com/products/big-ip-services/big-ip-dns, May 2021.

[40] Introducing FRED - fred. https://fred.nic.cz/en/, May 2021.

[41] Google. google/nomulus: Top-level domain name registry service on Google App Engine. https://github.com/google/nomulus/, May 2021.

[42] CZ.NIC. src · master · fred / cdnskey-scanner · GitLab. https://gitlab.nic.cz/fred/cdnskey-scanner/-/tree/master/src, Apr. 2021.

[43] RIPE NCC. GitHub - RIPE-NCC/rcdss: RIPE NCC CDS Scanner. https://github.com/RIPE-NCC/rcdss/, June 2021.

[44] PowerDNS. Add CDS publishing support · PowerDNS/pdns@ef54222. https://github.com/PowerDNS/pdns/commit/ef542223c831c0f791fb4909c4f5d06ed67398da, Oct. 2015.

[45] PowerDNS. Add CDNSKEY support · PowerDNS/pdns@088370c. https://github.com/PowerDNS/pdns/commit/088370cd587096453fdb81afe16c90e8d364a33f, Oct. 2015.

[46] Internet Systems Consortium. 4252. [func] Add

support for automating the generation CDS... · isc-projects/bind9@e939674. https://github.com/isc-projects/bind9/commit/e939674d53a127ddeeaf4b41fd72933f0b493308, Nov. 2015.

[47] CZ.NIC. dnssec: added CDS and CDNSKEY creation · CZ-NIC/knot@0d8dadb. https://github.com/CZ-NIC/knot/commit/0d8dadb47c10c79368187abf8f14b0c0cfe70ee2, May 2017.

[48] CZ.NIC. ksk rollover: parent DS check implemented +test · CZ-NIC/knot@3fbf23d. https://github.com/CZ-NIC/knot/commit/3fbf23dee007803e9a235ba09d21c5b5955bfde1, May 2017.

[49] Internet Systems Consortium. [master] dnssec-cds · isc-projects/bind9@ba37674. https://github.com/isc-projects/bind9/commit/ba37674d038cd34d0204bba105c98059f141e31e, Oct. 2017.

[50] CZ.NIC. Added AKM (reviewed) (bea9cd29) · Commits · fred / docs. https://gitlab.nic.cz/fred/docs/-/commit/bea9cd29bbabdc84b07ec9785ce8e487b6bd1fd3, Nov. 2017.

[51] Internet Systems Consortium. Introduce dnssec-policy configuration · isc-projects/bind9@a50d707. https://github.com/isc-projects/bind9/commit/a50d707fdcf702e9e07a3ad5ea74761cb6e3d08d, Nov. 2019.

[52] RIPE NCC. README update, add LICENSE · RIPE-NCC/rcdss@8b81bc6 · GitHub. https://github.com/RIPE-NCC/rcdss/commit/8b81bc6e1ccc48ee799422d47642540df6f84b6c, June 2021.

[53] Internet Systems Consortium. Manual Pages — BIND 9 documentation. https://bind9.readthedocs.io/en/v9_16_17/manpages.html, June 2021.

[54] PowerDNS. CDS/CDNSKEY acceptance · Issue #2756 · PowerDNS/pdns. https://github.com/PowerDNS/pdns/issues/2756, Sept. 2015.

[55] CZ.NIC. 8. Automated keyset management — FRED 2.42 Documentation. https://fred.nic.cz/documentation/html/Concepts/AKM.html, June 2021.

[56] CZ.NIC. fred-akm.conf.example · master · fred / akm · GitLab. https://gitlab.nic.cz/fred/akm/-/blob/master/fred-akm.conf.example, July 2017.

[57] M. Shchavleva. Architecture for monitoring CDNSKEY records from multiple sites for automated DNSSEC management in FRED system. Nov. 2019.

[58] M. Kerrisk. cron(8) — Linux manual page. https://man7.org/linux/man-pages/man8/cron.8.html, Sept. 2013.

[59] F5. ltm dns dnssec zone. https://clouddocs.f5.com/cli/tmsh-reference/v15/modules/ltm/ltm_dns_dnssec_zone.html, Sept. 2018.

[60] F5. AskF5 | Manual Chapter: Configuring DNSSEC. https://techdocs.f5.com/en-us/bigip-15-1-0/big-ip-dns-services-implementations/configuring-dnssec.html, June 2021.

[61] CZ.NIC. Configuration — Knot DNS 3.0.7 documentation. https://www.knot-dns.cz/docs/3.0/html/configuration.html#automatic-dnssec-signing, June 2021.

[62] Internet Systems Consortium. DNSSEC Guide — BIND 9 documentation. https://bind9.readthedocs.io/en/latest/dnssec-guide.html, June 2021.

[63] SWITCH. Automated DNSSEC Provisioning Guidelines for CDS processing at SWITCH. https://www.nic.ch/export/shared/.content/files/SWITCH_CDS_Manual_en.pdf, Mar. 2021.

[64] CZ.NIC. CZ.NIC - Automatizovaná správa keysetů. https://www.nic.cz/page/3909/automatizovana-sprava-keysetu/, May 2021.

[65] SK-NIC. SK-NIC has launched a technological innovation as the third country in Europe! – SK-NIC. https://sk-nic.sk/sk-nic-has-launched-a-technological/, May 2021.

[66] Internetstiftelsen. Making the internet safer - DNS-Labs - The Swedish Internet Foundation. https://internetstiftelsen.se/dns-labs/projects/, May 2021.

[67] .PT. Plano de Atividadese Orçamento 2021. https://www.dns.pt/fotos/editor2/relatorios/pa_2021.pdf, May 2021.

[68] SWITCH. CDS Status Check - Security - Internet Domains. https://www.nic.ch/security/cds/index.html, June 2021.

[69] CZ.NIC. Ticket #30847 - Fix case - multiple cdnskeys in different order on nameservers (ff868b8c) · Commits · fred / akm. https://gitlab.nic.cz/fred/akm/-/commit/ff868b8ceec2cad5c652adcac56b16f3af8106f0, June 2021.

[70] A. Buddhdev. RIPE NCC DNS Update. https://ripe82.ripe.net/wp-content/uploads/presentations/70-RIPE82_DNS_Update.pdf, May 2021.

[71] O. Caletka. CDS scanning at RIPE NCC. https://71.schedule.icann.org/meetings/vv7XkuePvghwFaLgt, June 2021.

[72] RIPE NCC. Reverse Delegation — RIPE Network Coordination Centre. https://www.ripe.net/manage-ips-and-asns/dns/reverse-dns, Sept. 2016.

[73] RIPE NCC. ENUM — RIPE Network Coordination Centre. https://www.ripe.net/manage-ips-and-asns/dns/enum, Apr. 2019.

[74] Nominet. Managing DNSSEC with EPP – Registrar Resources. https://registrars.nominet.uk/uk-namespace/registration-and-domain-management/dnssec/managing-dnssec-with-epp/, June 2021.

[75] S. Crocker and S. Huque. DS Updates and Multi-Signer Coordination – A Continuing Series ICANN 71, ”The Hague” – Episode 5. https://71.schedule.icann.org/meetings/vv7XkuePvghwFaLgt, June 2021.

[76] Cloudflare. Expanding DNSSEC Adoption. https://blog.cloudflare.com/automatically-provision-and-maintain-dnssec/, Sept. 2018.

[77] ICANN. ICANN71 | DNSSEC and Security Workshop (2 of 2). `https://71.schedule.icann.org/meetings/vv7XkuePvghwFaLgt`, June 2021.

APPENDIX A

A list of analyzed TLDs, ordered by number of domains according to zonefiles.io [22]

.com	.jp	.kr	.cl
.net	.app	.nu	.рф
.uk	.网址	.dev	.tk
.org	.ca	.buzz	.world
.de	.icu	.tw	.monster
.ru	.be	.life	.design
.info	.dk	.ro	.link
.xyz	.work	.cloud	.id
.ch	.in	.fi	.page
.nl	.store	.gr	.il
.online	.cz	.no	.art
.cn	.es	.fun	.pt
.it	.live	.asia	.email
.fr	.at	.nz	.ltd
.se	.co	.us	.cat
.biz	.ir	.blog	.me
.top	.tech	.mx	.su
.eu	.mobi	.ar	.today
.site	.space	.vn	.ie
.club	.website	.tr	.cyou
.pl	.ua	.sk	.digital
.br	.hu	.tokyo	.one
.au	.za	.bar	.xxx
.shop	.pro	.ga	.group
.vip	.io	.tokyo	.solutions