Check-Repeat: A New Method of Measuring DNSSEC Validating Resolvers

Yingdi Yu
UCLA
yingdi@cs.ucla.edu

Duane Wessels
Verisign Labs
dwessels@verisign.com

Matt Larson
Verisign Labs
mlarson@verisign.com

Lixia Zhang
UCLA
lixia@cs.ucla.edu

Abstract—**As more and more authority DNS servers turn on DNS security extensions (DNSSEC), it becomes increasingly important to understand whether, and how many, DNS resolvers perform DNSSEC validation. In this paper we present a query-based measurement method, called Check-Repeat, to gauge the presence of DNSSEC validating resolvers. Utilizing the fact that most validating resolver implementations retry DNS queries with a different authority server if they receive a bad DNS response, Check-Repeat can identify validating resolvers by removing the signatures from regular DNS responses and observing whether a resolver retries DNS queries. We tested Check-Repeat in different scenarios and our results showed that Check-Repeat can identify validating resolvers with a low error rate. We also cross-checked our measurement results with DNS query logs from .COM and .NET domains, and confirmed that the resolvers measured in our study can account for more than 60% of DNS queries in the Internet.**

I. Introduction

Domain Name System Security Extensions (DNSSEC) provide the much needed cryptographic protection for the critical DNS services, by allowing end hosts to authenticate DNS data. Effective rollout of DNSSEC requires deployment efforts from both data publishers (zone owners) and data consumers (DNS clients). Zone owners must sign their zones and publish their keys, and DNS clients must upgrade their caching resolvers to perform cryptographic verification of the DNS data.

Tracking both sides of DNSSEC deployment is important for a number of reasons. For example, it helps with the "chicken-and-egg" problem. When publishers know that consumers are configuring validation in their resolvers, they will be increasingly motivated to sign their zones. When consumers know that publishers are increasingly signing their zones, they will be more motivated to enable validation.

The publishing side of DNSSEC deployment has been well studied over the last few years [1]. It is relatively easy to determine whether a given zone is signed by simply sending queries to the authoritative DNS servers of the zone. In contrast, it is rather difficult to measure how many caching resolvers have turned on DNSSEC verification. Generally speaking one does not know all the existing caching resolvers around the world, nor can one directly query them externally. To get good measurement results, one must first find a way to capture resolver queries and analyze their behavior.

We are particularly interested in understanding DNSSEC deployment at the consumer-side. To gauge the number of caching resolvers that perform DNSSEC validation, which we call *DNSSEC validators*, or validators in short, we face the following two problems:

- How can one gather DNS queries from as many caching resolvers as possible?
- For a given resolver, how can one tell whether it performs validation of DNS data?

Some previous studies solved the first problem by analyzing the queries received by the authority DNS servers for a popular domain, such as .ORG [2] and .JP [3]. Unfortunately this method can potentially introduce false positive results. As reported by [4], about 1.6% of measured caching resolvers sent DNSKEY query, but did not validate data.

Some other studies solved the second problem by combining the observations of end host behavior on both DNS queries and HTTP requests [4], [5]. As we discuss in Section VI, the effectiveness of this approach is confined to the resolvers that query for a specific domain name. Ideally, one wishes to derive a method that can address both problems simultaneously while avoiding the above mentioned negative factors.

In this work, we develop a new solution, dubbed Check-Repeat, to address the two problems at once. Our earlier measurements show that most validating resolver implementations (such as BIND and Unbound), upon receiving a bad response [6], will resend a query to another authority server. Utilizing this observation, Check-Repeat first redirects DNS queries from multiple zones to a single, experimental zone named VALIDATORSEARCH.VERISIGNLABS.COM, intentionally removes the DNSSEC signature records from responses given by that zone, and then observes whether the caching resolvers will resend queries to another server of VALIDATORSEARCH.VERISIGNLABS.COM zone. We examine whether a resolver sends both DNSKEY queries and repeated queries as a stronger indicator of performing DNSSEC validation. Section III provides the full details of Check-Repeat's operations.

We also compared our measurement results with previous studies, and found that the ratio of validators and resolvers measured are consistent. We also evaluated the representativeness of our results by cross-checking them against query logs from the .COM and .NET authority servers. We observed that the set of caching resolvers seen in our measurements accounted for more than 60% of queries to .COM and .NET zones. The full results of our analysis are presented in Section IV.

In the rest of the paper, we first briefly introduce the basic concepts of DNS and DNSSEC in Section II. We then proceed to describe our methodology and measurement results (Sections III and IV). We discuss the deployment issues surrounding our measurement tool in Section V and related work in Section VI. We conclude the paper in Section VII.

II. BACKGROUND

A. DNS

DNS provides mappings from domain names (e.g., WWW.UCLA.EDU) to IP addresses, and a wide range of other data, such as E-mail servers and address-to-name records. All of these mappings are represented in terms of Resource Records (RRs). For example, "www.ucla.edu A 169.232.33.241" is an A type RR that maps the name WWW.UCLA.EDU to its IP address, and "www.verisign.com CNAME www-ilg.verisign.net" is a CNAME type RR that redirects the resolution of WWW.VERISIGN.COM to the resolution of another domain name. All resource records in a zone (e.g., UCLA.EDU) are stored in one or more servers called *authority servers*, which can answer queries for domain names in that zone.

When an end user wants to know the IP address of a domain name, it usually asks a special name server, called a *caching resolver*. Usually, the caching resolver sends queries directly to authority servers. In some situations, however, a caching resolver might be configured to *forward* its queries to another caching resolver. Sometimes resolvers send queries for names that do not exist [7]. These domain names are called NXDOMAIN names.

B. DNSSEC

When the DNS was initially designed, data authentication was not taken into consideration. For example, when a resolver receives an RRset of the IP addresses of the name WWW.UCLA.EDU, the resolver can not tell if this RRset is created by the operator of UCLA.EDU zone or by a malicious attacker.

DNSSEC is an extension to the DNS, providing a solution to data authentication. A DNSSEC-secured zone generates at least one pair of public/private keys. Private keys are used to sign each RRset in the zone. Such a process is called *zone signing*. The generated signatures are added to the zone data as RRSIG records. When responding to a DNSSEC-capable resolver, the authority server attaches the signatures in the reply message.

Validating resolvers use the zone's public key to verify that the signatures are authentic. Whereas private keys must be kept in secret, public keys are published in the zone file, for anyone to retrieve, as DNSKEY records. Validating resolvers are expected to issue separate queries for the public keys as necessary.

Although responses can be validated by verifying attached signatures, DNS is still not secured without authenticating public keys, i.e., which public key should be used to verify signatures? To solve this problem, a parent zone can point to a public key in each of its signed child zones. Such information is stored in one or more Delegation Signer (DS) records. A zone's DS record is signed and held by its parent zone, which means it can be validated by the parent zone's public key. The public key pointed by a DS record is called Key Signing Key, which is only used to sign the DNSKEY RRset in a zone. The other DNSKEY records (called Zone Signing Key) are used to sign the other RRsets. To validate a response, a resolver needs to check 1) whether the attached signature can be verified with the corresponding public key; 2) whether the public key can be validated with the DS record signed by the parent zone. Such a verification process continues recursively back along the zone's ancestors (as shown by the dotted line in Figure 1) until reaching a trusted public key that has been configured locally as a trust anchor. The whole process is called *DNSSEC validation*, and a resolver that can validate DNSSEC data is called a *DNSSEC validator*. Errors at any step in the process of validation can cause name resolution failures.

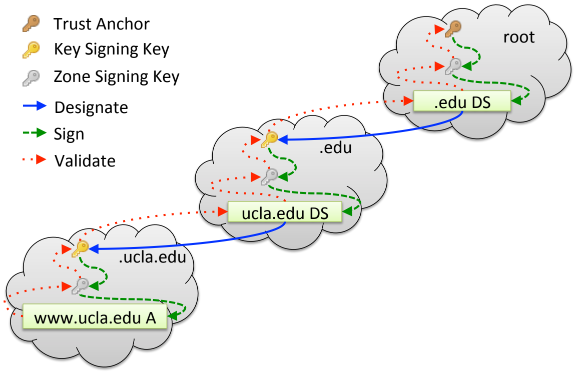

Fig. 1. An example of how DNSSEC validation works.

III. METHODOLOGY

In this section, we explain how Check-Repeat functions to measure DNSSEC validators. Check-Repeat consists of three components: a set of CNAME records that are used to collect queries from caching resolvers, a signature remover that can induce validators to retry queries, and a query-based validator identifying algorithm.

A. Validation Indicators

Although receiving DNSKEY and DS records are necessary steps in DNSSEC validation, they are not always sufficient indicators that validation is actually taking place. DNSKEY and/or DS queries may be sent by monitoring systems, browser plugins, DNS crawlers, and even end users. Another indicator of validation is the rejection of bad data, such as missing or incorrect signatures, keys, or DS records. Since authority servers always do their best to provide good data, we won't find evidence of rejection in the query logs of operational zones. To elicit this behavior, we must intentionally introduce errors into the communication between validators and signed zones.

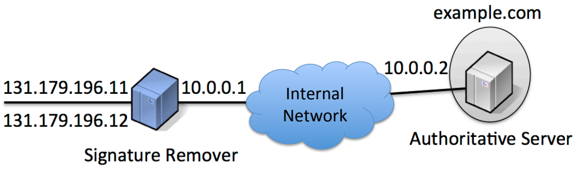

Fig. 2. An example of deploying signature remover in front of a zone. 131.179.196.11 and 131.179.196.12 are the original IP addresses of the two authority servers respectively.

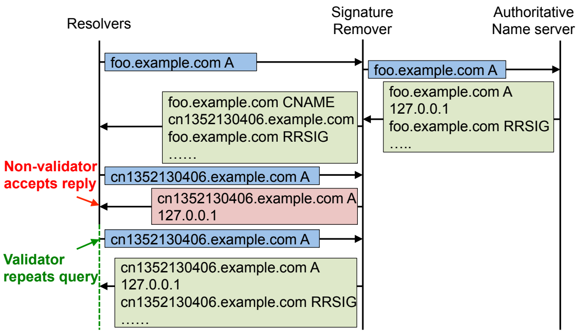

Fig. 3. An example of how signature remover works.

We built a tool called *Signature Remover* which can intercept queries to a zone and remove signatures from specific responses. In many cases, the missing signature causes the validator to retry its query at one of the zone's other authority servers. Non-validators, however, accept the modified response as legitimate. Therefore, the repeated queries can be used as a strong indicator of DNSSEC validation. Since the signature remover modifies signature records only, it does not affect resolution for non-validators. Note that we do not use repeated queries alone as an indicator, since repeats might also be caused by packet loss or other reasons.

Figure 2 shows how a signature remover is placed in front of a signed zone, EXAMPLE.COM. In this case, the zone has two authority servers and a single signature remover process listens on both[1]. Later, in Section V we talk about how the signature remover can be used in cases where the authority server are distributed. The signature remover forwards certain queries to the real ("backend") authority server using loopback or private IP addresses.

With such a deployment, the signature remover receives all DNS traffic for the EXAMPLE.COM zone. However, it doesn't necessarily remove signatures for every name and record in the zone. It is configured with a *target set* of names. Queries for names that are not in the target set are passed on directly to the backend server, and the corresponding responses are passed back to the querier unmodified. Furthermore, we restrict the types of records for which signatures are removed (i.e., A and AAAA).

Queries for names in the target set are called *target queries*. Figure 3 shows how a signature remover handles a target query for the address of FOO.EXAMPLE.COM:

1) When the signature remover receives a target query, it responds with a CNAME record. The CNAME record contains a unique component, based on the query time, such as CN1352130406660979.EXAMPLE.COM. At the same time, the signature remover pre-fetches the response for the target query from the backend name server.
2) A resolver receiving a response containing the CNAME record will send another query for the new name. Such a query is called a *probe query*. The probe query will be intercepted by the signature remover as well.
3) The first time that the signature remover receives a particular probe query, it returns an unsigned response which is constructed using the RRs in the answer section of the pre-fetched response.
4) A non-validator will not notice the missing signatures, thus accepting the unsigned response. The signature remover will not receive any more probe queries.
5) A validator, due to validation failure, can send the probe query to another authority server address, which is intercepted by the signature remover again.
6) When the signature remover sees a repeated probe query, it returns a signed response which can be validated.

With the signature remover deployed, we define a *trial* as a sequence of queries containing a target query and its subsequent probe queries. The basic query pattern of a validator consists of a target query and two probe queries. Depending on whether a validator has previously fetched the zone's public key, the pattern may also include a DNSKEY query. Note that different implementations may request records in a different order, or may include other queries in the sequence. However, to be considered as a validator, it must contain the basic pattern as a subset. A log file records all queries seen by the signature remover. We analyze this log file to identify validators using the methods described in Section III-C.

B. Sample Resolver Collection

For our measurements to be meaningful, we need to receive DNS queries from many resolvers spread far and wide throughout the Internet. Since removing signatures from DNS responses might negatively impact end users (i.e., with SERVFAIL errors), we must carefully choose the zones on which to deploy the signature remover. We want to find zones that are both busy, yet won't cause disruptions when we tinker with them. We need zones where the answers don't really matter. Fortunately, we happened to have some.

In our measurement, we managed to add CNAME records for five "WPAD" domain names, WPAD.{COM, NET, ORG, BIZ, US}, so that queries for these names will eventually become queries to our experimental zone. The Web Proxy Auto Discovery protocol (WPAD) is a way for browsers to locate proxy auto-configuration scripts using DNS queries. For example, a web browser may query for WPAD.CS.UCLA.EDU to get the location of a local proxy configuration file. If there is no such a file or the location does not exist, the web browser

[1] Although putting all authority servers on one network violates the DNS best practices, we feel it is acceptable for this study.

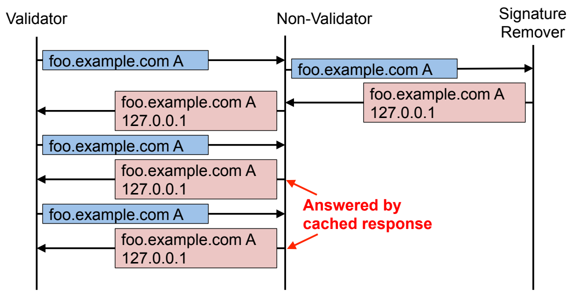

Fig. 4. An example of how multi-level cache prevent repeated queries from reaching signature remover.

TABLE I
SUMMARY OF CAPTURED DNS QUERY TRACE

	trace-I	trace-R
Captured Queries	24,786,845	8,030,734
Collected IPs	77,685	50,603
Days	36	9

TABLE II
SUMMARY OF MEASUREMENT RESULT OVER TRACE-I

	Total	Validators	Ratio
Trials	6,498,277	561,772	8.64%
IPs	77,685	2,768	3.56%
Resolvers	49,488	2,377	4.80%

may query for WPAD.UCLA.EDU instead. It does not make any sense to query for WPAD.EDU if the web browser cannot find the configuration file yet. However, many web browsers (especially Microsoft Internet Explorer) still query for these meaningless domain names. And the five domain names we used receive millions of queries per day that, theoretically, should not exist.

C. Query Log Analysis

Within the signature remover's log file, a resolver is identified as a validator if it satisfies the following three rules:

A. it received a signature via a repeated query;
B. it received the zone's public key via a DNSKEY query;
C. its query pattern was consistent over time.

The first two rules are necessary steps of DNSSEC validation. The purpose of the last rule is to exclude two situations that can cause false positive results: 1) non-validators may occasionally re-send queries due to packet loss, and 2) non-validators and validators might coexist behind a single NAT box.

It is important to note that, based on the above rules, our method cannot identify validators that forward their queries to another caching resolver. As shown in Figure 4, the non-validator is not aware of the missing signature and caches the response until it is ejected out. During this time, even if the validator repeats the probe query, the non-validating forwarder responds with the cached data, rather than contacting the authority server again. Such multi-level caches can cause false negatives in our measurements.

The first step of query analysis is to group queries in the same trial together. Since all queries in a trial are associated with a unique CNAME record, they can be grouped according to this information. Any queries arriving more than five seconds after the target query are discarded. If we do not find a repeated probe query within five seconds of the timestamp, the trial is considered as a non-validation.

Next, for each trial, we determine whether or not the resolver requested the public key, or if the key was already in the resolver's cache. This is somewhat tricky because the key is not unique per-trial and has a relatively long TTL (two hours). Furthermore, some trials issue queries from multiple IP addresses. Therefore, we first group together all the IP addresses for a resolver (within a 24-hour period). Then we can see if any of those IP addresses received a DNSKEY response within two hours of the trial. If we are not able to locate such a DNSKEY query, the trial is considered as a non-validation.

Finally, we categorize all IP as validating, non-validating, or mixed. IP addresses that are only associated with non-validating trials are marked as non-validators. Similarly, addresses that are only associated with validating trials are marked as validators. The remaining set of addresses participated in both validating and non-validating trials. Here we use a threshold. If 90% or more of an address' trials are non-validating, the address is marked as a non-validator. The remaining are most likely IP addresses of NAT boxes.

IV. MEASUREMENT RESULT

In this section, we examine our measurement results. We placed a signature remover in front of an experimental zone VALIDATORSEARCH.VERISIGNLABS.COM. We have only one name in the target set. The five "WPAD" domain names, WPAD.{COM, NET, ORG, BIZ, US}, are all configured with a CNAME record pointing to the target name. We collected queries from 2012-09-25 to 2012-10-31, and refer to this signature remover data as *trace-I*. Then we disabled the signature remover, and collected queries from 2012-11-01 to 2012-11-09 in order to check for false positives. We refer to the data in this latter period as *trace-R*. Table I summarizes these two traces.

Trace-I contains 6,498,277 trials from from 77,685 distinct IP addresses. Upon applying the validation detection techniques described in Section III, we find 561,772 trials classified as DNSSEC validations. These validation trials consist of queries from 2,768 distinct IP addresses. We grouped IP addresses belonging to the same resolvers together, thus getting 49,488 resolvers and 2,377 validators. We list the summary of validator identified over trace-I in Table II. The ratio of validators to total resolvers is 4.8%, which is close to the 4.5% measured by Wander and Weis [4] and 4.0% measured by Huston [5] using the rejection-of-bad-data technique.

In order to check false positive rate, we also applied our validator identifying algorithm to trace-R. Since the signature remover was turned off when collecting queries in trace-R, trace-R should not contain repeated queries caused by

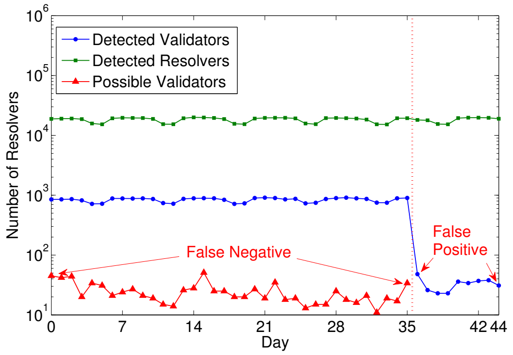

Fig. 5. Resolvers and validators measured from 2012-09-25 (Day 0) to 2012-11-09 (Day 44). On 2012-11-01 (Day 35) signature remover was turned off.

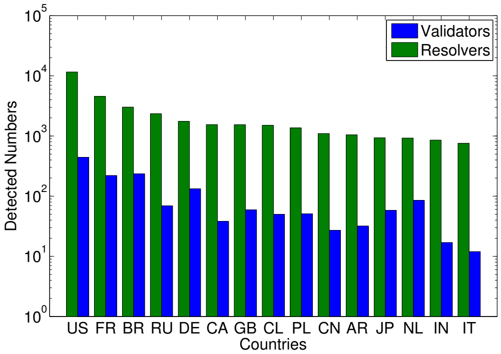

Fig. 6. Geo-distribution of detected resolvers and validators by countries

intentionally missing signatures. If Check-Repeat still reported validators from trace-R, then this implies that Check-Repeat falsely identified a resolver as a validator by mistaking repeated queries caused by other reasons (e.g., packet loss). All of these identified "validators" are false positives.

Figure 5 shows the number of validators and resolvers detected each day for both traces. Because the signature remover was turned off on 2012-11-01 (i.e., Day 36), any validators identified since that time are false positives. As shown in Figure 5, the number of identified validators has dropped to around 40 since Day 36. Given the number of resolvers found per day is approximately 20,000, the rate of false positive detection is only 0.2%. This suggests that Check-Repeat introduce very few false positives.

To evaluate the false negative rate of Check-Repeat, we examined trace-I for resolvers that received the public key, but did not repeat queries. These resolvers might be validators that can not be identified by Check-Repeat. For example, a validator may terminate a trial as long as one response is not signed, or the repeated query is answered by an up-level caching resolver. As the red line shows in Figure 5, the daily false negative count is around 30. Given that the number of caching resolvers found per day is approximately 20,000, the false negative rate is about 0.15%. We believe that most false positives are due to multi-level caches, a.k.a. DNS-forwarding.

To evaluate the representativeness of our measurement results, we cross-checked them against the query logs of G.GTLD-SERVERS.NET (G.GTLD for short) which is one of the 13 authoritative servers of .COM and .NET. We find that, although IP addresses measured in our results comprise only 1.6% of the addresses seen by G.GTLD, those addresses account for 63.5% of all the queries to G.GTLD. This implies that our measurement results can reflect the validation ratio of considerable portion of DNS traffic in the Internet.

Among the 63.5% of G.GTLD's queries sent from caching resolvers seen in our measurements, 80.6% are sent by non-validators, and 19.4% by validators. Surprisingly, even though our validators account for only 0.056% of the addresses seen by G.TLD, they account for 12.3% of the queries. Such a fact suggests that validation has been configured on some very large or busy caching resolvers. Not surprisingly, Comcast, who has been very active and outspoken on DNSSEC validation [8], is a significant contributor to validating queries. In fact, 73.4% of DNSSEC-secured responses are sent to Comcast's resolvers.

Next, we examined the geographic distribution of resolvers and validators collected in our data. The identified resolvers are distributed among 219 countries, 115 of which have at least one validator. Figure 6 shows the number of resolvers and validators found for the top 15 countries. We also ranked countries that have at least 200 identified resolvers by the ratio of validators to resolvers. We find that Sweden, Czech Republic, and Finland have the highest ratio. This is not surprising as each of these countries also lead in adoption of DNSSEC within their ccTLD. The countries with the lowest ratios, such as Korea and Thailand, are all in Eastern Asia.

As mentioned in Section III, we also find that different resolver implementations have unique query patterns when validating. This provides us with a chance to estimate the current market share among validators. We find that 60% of detected validators are using BIND, 29% are probably Unbound, and 9% of are running Nominum's software.

V. Discussion

In this section, we first take a look of the measurement results and then discuss how other parties may deploy Check-Repeat in their networks.

A. *Gauging the Deployment of DNSSEC Validators*

Gudmundsson and Crocker [2] performed DNSSEC validator measurement one year ago. Their results show that about 10% of queries to .ORG domain are sent by validators. Our results showed that more than 12% of queries to .COM and .NET domains are sent by validators. The number of DNS queries sent by validators seems growing over time, albeit at a rather slow pace. However, we would point out that the results reported in [2] may contain many false positives, so that the actual percentage of queries from validators back then may have been lower than 10%. Thus the DNSSEC validator numbers may be growing faster than the data suggests.

B. A Distributed Signature Remover

In this measurement, we deployed only one signature remover in front of the authority server of VALIDATORSEARCH.VERISIGNLABS.COM zone. However for others to conduct the Check-Repeat measurement study, it would be infeasible to assume that all the authority servers for a zone are placed within a single network. Real zones are served from multiple networks and multiple locations. Thus one needs to run the signature remover for a distributed zone.

However, deploying distributed signature removers may make it difficult to decide when to remove signatures or to identify repeated queries. Rather, it is possible that none of the responses will include the necessary signature, in which case the resolver returns a SERVFAIL error code to its client.

Knowing this, we can choose to deploy signature remover only for domains that can tolerate a SERVFAIL error. To many applications, NXDOMAIN and SERVFAIL errors are identical. They both mean "host not found." Thus, it may be acceptable to use this technique on names that normally return NXDOMAIN.

C. Extending the Range of Measurement

Unlike other approaches, our technique works purely over DNS. It does not require application-level transactions, such as HTTP. While the data presented in this paper did come from browser-initiated queries, we'd like to be able to extend our measurement footprint by including other types of domains, such as IN-ADDR.ARPA reverse zones, and those used in the process of mail delivery, such as defunct realtime blackhole lists (RBLs).

VI. Related Work

Gudmundsson and Crocker [2] performed measurement to identify DNSSEC validators by examining query logs from the .ORG domain. They counted all the resolvers who sent DNSKEY and DS queries as DNSSEC validators. Given the popularity of .ORG domain, their measurement results should capture most if not all the caching resolvers. However although fetching DNSKEY and DS records is necessary for performing DNSSEC validation, it is not a sufficient indicator and can introduce false positives. Our own measurement results and other previous work [4] show that many resolvers query for DNSKEY record but do not perform signature validation. Moreover, since .ORG's authoritative servers use anycast, it is difficult to assemble a complete trace of all the queries from a given resolver. Fujiwara [3] used a similar method to identify DNSSEC validators by analyzing query logs from the .JP domain.

Huston [5] and Wander & Torben [4] looked for the absence of the use of intentionally bad data to detect a DNSSEC validator. For example, one can embed in a web page two images with different URLs and different hostnames. One hostname exists in a properly signed zone, while the other is incorrectly signed. When the user agent attempts to load both images, one can observe whether the involved DNS resolver is configured to perform DNSSEC validation. If the user agent loads the good image, but not the bad one, this shows a strong indication that the resolver performed validation. These studies also use unique-per-trial query names and require correlation of DNS and web server log files. To maximize the measurement size, [5] utilized paid advertisement, while [4] used a combination of voluntary website visitors and hidden HTML/javascript added by webmasters. While this technique has the advantage of "seeing through" forwarders, it also has the disadvantage of identifying specific non-validating forwarders as validators.

Measurement efforts on the deployment of DNSSEC at the publisher side is relatively easier with better established results. Osterweil *et al.* proposed three metrics to measure this deployment: availability, verifiability, and validity [9]. They also built a monitoring system called SecSpider [1] to track the publisher side of DNSSEC deployment.

VII. Conclusion

In this paper, we proposed Check-Repeat, a new query-based method to measure DNSSEC validators. This method leverages the knowledge that, upon receiving an intentionally unverifiable DNS response, most caching resolvers that perform DNSSEC validation will repeat their queries, and uses the presence of repeated queries as a strong indicator of DNSSEC validation. Check-Repeat combines the advantages of previous work. It achieves the same level of accuracy as the methods used by other researchers and allows us to collect measurement results from a sufficient number of caching resolvers. By cross-checking our results with data from the .COM and .NET authority servers, we learn that Check-Repeat finds the resolvers that are responsible for producing more than 60% of DNS query traffic to .COM/.NET, and that about 12% of .COM/.NET responses are currently sent to validating resolvers.

References

[1] "SecSpider." [Online]. Available: http://secspider.cs.ucla.edu/

[2] O. Gudmundsson and S. Crocker, "Observing DNSSEC Validation in the Wild," *Securing and Trusting Internet Names (SATIN)*, 2011.

[3] K. Fujiwara, "Number of DNSSEC validators seen at JP," in *ICANN 41*, 2011.

[4] M. Wander and T. Weis, "Measuring Occurrence of DNSSEC Validation," in *PAM*, 2013.

[5] G. Huston, "Counting DNSSEC," RIPE Labs website. [Online]. Available: https://labs.ripe.net/Members/gih/counting-dnssec

[6] "dnssec-or-not.net." [Online]. Available: http://dnssec-or-not.net/

[7] D. Wessels and M. Fomenkov, "Wow, thatsa lot of packets," in *Proceedings of Passive and Active Measurement Workshop (PAM)*, 2003.

[8] "Comcast's Operational Experiences," in *ICANN DNSSEC Workshop*, March 2012. [Online]. Available: http://costarica43.icann.org/meetings/sanjose2012/presentation-comcasts-operational-experiences-14mar12-en.pdf

[9] E. Osterweil, M. Ryan, D. Massey, and L. Zhang, "Quantifying the operational status of the dnssec deployment," in *Proceedings of the 8th Internet Measurement ACM Conference*, 2008, pp. 231–242.