The Current State of DNS Resolvers and RPKI Protection

Marius Brouwer
University of Amsterdam
marius.brouwer@os3.nl

Erik Dekker
University of Amsterdam
erik.dekker@os3.nl

ABSTRACT

The goal of this research was to gain insight into the Resource Public Key Infrastructure (RPKI) protection state of DNS resolvers. RIPE Atlas Probes were used to send DNS queries to an authoritative DNS server. This server contained Resource Records in both an RPKI valid and invalid prefix. The RIPE Atlas probes were instructed to send their queries to the valid prefix through their configured DNS resolvers, which in turn were answered by a CNAME referencing to the invalid prefix. This enabled us to determine whether a probe's DNS resolver was RPKI protected or not. Our results show that on January 23rd 2020, 7% of the probes configured DNS resolvers were protected, this increased to 15% on February 3rd. Of these probes 11.5% was fully protected, due to having multiple DNS resolvers configured.

1 INTRODUCTION

The Border Gateway Protocol (BGP) was specified before security was of concern and abuse was not as prevalent as it is today [1]. In the original BGP specification, there were no security measures defined to prevent either intentional or unintentional network configuration errors [2]. The lack of native security measures makes BGP prone to both IP prefix hijacking (from now on referred to as "prefix hijacking") and route leaks [2]. Prefix hijacking is the phenomenon in which an Autonomous System (AS) maliciously announces itself as the origin of a prefix, whereas route leaks announces itself as being the shortest path to the origin prefix. These illegitimate route advertisements pollute BGP routing tables, and affect the confidentiality, integrity and availability of IP communication [3].

A recent example of this behavior, which had a significant impact on Internet traffic, was the BGP leak of DQE (AS33154) through Verizon (AS701) in June 2019 [4]. The ISP DQE was using a BGP optimizer that created more specific routes within its network for traffic engineering purposes. These routes were announced to their customer Allegheny (AS396541), which were then sent on to Allegheny's transit provider Verizon. Verizon proceeded to announce these more specific routes to their BGP peers, which routed all traffic to these more specific routes through Verizon, Alleghany and then on to DQE. This route leak incident lasted almost two hours and redirected all of this traffic through their networks causing service disruption. Manual intervention was required by DQE to resolve the issue.

To defend against these threats and secure inter domain routing, two main solutions have been proposed by the Internet Engineering Task Force (IETF), namely, the Resource Public Key Infrastructure (RPKI) and BGPsec [5, 6]. RPKI allows network operators to cryptographically sign and validate prefix origin data; additionally, BGPsec signs and validates the whole AS path. While BGPsec provides more security, it is not broadly adopted [7, 8]. For this reason, this paper will focus on RPKI.

Due to the distributed nature of BGP and RPKI, the majority of network operators should sign their network prefixes and implement RPKI filtering to minimize prefix hijacks and route leaks [9]. A study conducted in 2019 claims that between 9.98% and 11.28% of the BGP announcements are verifiable using RPKI [10].

Non-RPKI filtered ASes could also benefit from RPKI when enough ASes have implemented RPKI filtering, e.g., when an AS lacks RPKI filtering, but one of its upstream ASes does not, the invalid prefix might still be filtered by one of its upstream ASes. However, there are still situations that one may indirectly fall victim to prefix hijacks even if their own AS is RPKI protected. A good example of this is the Amazon Route 53 BGP hijack [11, 12]. In this situation, the prefixes of the authoritative Amazon Domain Name System (DNS) servers were hijacked. Any AS with a DNS resolver without RPKI filtering would receive a valid but malicious response from the hijacked authoritative DNS server, even if the AS where the query originated from was RPKI protected. It is not unusual that DNS resolving is done within another AS, such as is the case with public DNS resolvers. We argue that the network in which the DNS resolver resides also needs to be RPKI filtered, otherwise the DNS resolution will not be protected from prefix hijacks.

2 RESEARCH QUESTIONS

As mentioned in the introduction, it is important that the network in which the DNS resolver resides is protected by RPKI filtering. Therefore, the goal of this research was to investigate the protection status of DNS resolvers from the users perspective. To investigate this topic, we proposed the following main research question:

" What is the state of RPKI filtering on DNS resolvers? "

Since public DNS resolvers also reside within networks that use anycast. We suspected that these DNS resolver providers, due to their more complex environment, might have more difficulty to protect themselves with full RPKI protection. Therefore, we proposed the following sub question:

How does anycast influence the level of RPKI protection on DNS resolvers?

RPKI benefits from so called herd immunity [9]. If enough backbone providers have RPKI filtering, their downstream ASes will benefit for this as well. This makes them indirectly protected even without filtering within their own AS. Therefore, we proposed to research the following sub question:

How does the length of the AS path between resolver and authoritative DNS server influence the level of RPKI protection?

3 BACKGROUND

This paper will go into detail about topics such as BGP, prefix hijacking and RPKI. To provide an overview of these topics we will explain them in this section.

3.1 BGP

BGP is the de-facto standard inter domain routing protocol used on the Internet [13]. In brief, BGP speakers can peer with other BGP speakers and announce prefixes through a series of ASes. While BGP is highly scalable, it was designed before the Internet became subject to attacks, which lead to routing vulnerabilities being exploited [2]. BGP has developed some options that can improve security, such as explicitly configuring BGP peers, deploying BGP session shared secrets and filtering on the peering interface amongst others [2]. However, because BGP is based on trust, it is still prone to several threats.

3.2 BGP Vulnerabilities and Attack Variations

A systemic vulnerability with BGP is that ASes can advertise learned routes beyond their intended scope or advertise prefixes that they are not authorized to [14]. The first event is known as a route leak and the latter as a prefix hijack. Both of these events could either be intentional or unintentional.

Research done by Birge-Lee et al. identified the following types of prefix hijacks [15]:

- Traditional sub-prefix hijack: announcing a more specific prefix than the actual AS owner, which attracts all traffic to the more specific on a global scale. This attack is effective on a global scale but also very obvious and highly suspicious since the "victim" has never announced the more specific prefix.
- Traditional equally-specific-prefix hijack: the prefix announced is the same length as the authorized AS. This will only attract traffic which would have a shorter path or local preference towards the hijacked AS. This attack is effective on a local scale and therefore less obvious than a sub-prefix attack.
- Prepended sub-prefix hijack: the attacker announces a more specific prefix that contains the address space he wants to hijack, and manipulates the route by prepending the victim's AS followed by the attacker's AS. From this point on, the traffic is not routed any further since there is no connectivity between the attacker's and victim's AS. This attack is effective on a global scale and much more effective than a traditional sub-prefix hijack.
- Prepended equally-specific-prefix hijack: this attack is similar to the prepended sub-prefix hijack, except for the fact that the announced prefix is just as long as the victim's prefix. It will not attract much traffic since BGP path selection will prefer the shorter path. The attackers path is now one hop longer due to prepending the victim's AS in the route.

An example of a route leak as presented by by Birge-Lee et al. [15]:

- AS-path poisoning attack: the attack uses BGP to man-in-the-middle traffic heading towards the victim. This is achieved by announcing a more specific route than the victim and appending the legitimate route to the victim following its own AS. One requirement is that the attacker needs to be multi-homed. One of the up-link providers is required to receive the traffic and the second to forward the traffic on towards the legitimate path.

When one of the previous attacks succeeds and a prefix is hijacked or leaked, a malicious AS can black hole or intercept the hijacked traffic and impersonate the legitimate receiver of the traffic [16]. Once a network is subject to prefix hijacks, the AS, authorized to announce the prefix, will start losing traffic. It can do very little about this except for directly contacting the leaking party to stop advertising these route leaks [17].

3.3 BGP Security Extensions

Since BGP does not provide any native security measures to remedy these threats, several extensions have been proposed to improve the security of BGP, namely, soBGP [18], S-BGP [19], BGPsec [20], Internet Routing Registries (IRR) [21] and RPKI [6]. The study of Lychev et al. has indicated that these network protocols offer only marginal benefits over RPKI and that transitioning to these protocols is expected to be slow [8]. Chung et al. identified that RPKI has seen a rapid increase in the recent years [10]. Of these solutions, RPKI is the most mature and gained the most traction, therefore we will provide more background on this solution.

3.4 RPKI

Before RPKI came to fruition, IRR was used to verify route origination. However, IRR has no mechanism that verifies if the registrant inputs correct information [9]. RPKI addresses the issue of erroneous data by using cryptographically signed certificates that connect the AS to authorized prefixes.

3.4.1 PKI. RPKI makes use of Public Key Infrastructure (PKI) to sign so called Route Origin Authorization (ROA) records. ROA records contain the association between AS and authorized prefixes to announce. Additionally, it may also contain the maximum length of the prefix, if not specified the AS is only authorized to announce the prefix specified and not a more specific prefix.

3.4.2 RPKI validity states. RPKI has three validity states, namely VALID, INVALID and UNKNOWN. When an RPKI route is VALID, this means that the route announcement is covered by at least one ROA. The INVALID state means that the prefix is announced by an unauthorized AS or the prefix is more specific than covered by the ROA. Last, an UNKNOWN state means that the prefix is not covered by a ROA.

A route will be evaluated as either VALID, INVALID or UNKNOWN once BGP Route Origin Validation (ROV), also referred to as RPKI filtering, takes place. With RPKI filtering, routes will only be dropped when a route announcement has the INVALID state.

PKI relies on a Trusted Third Party (TTP) to sign and publish the public key. The TTP is typically known as the Certificate Authority (CA). In RPKI terminology this is referred to as the Trust Anchor (TA). RPKI relies on the five Regional Internet Registries (RIR) to function as TAs, being: AFRINIC, APNIC, ARIN, LAPNIC and RIPE NCC. These organizations maintain the allocation and registration of IP addresses within the specific world regions. As such they are

able to verify the association between prefixes and AS contained in the signed ROA.

RPKI solves the problem of ASes INVALID routes propagating once filtering has been enabled. If traffic is directed towards a route that has been leaked or hijacked it will traverse multiple ASes to reach a specific IP. Suppose the traffic originates from AS100 and traverses AS200 and AS300, to reach an invalid prefix announced at AS400. If any of the ASes has RPKI filtering on the path to AS400, the route will be dropped and therefore protect the traffic from being intercepted by AS400 since it is not authorized to announce this prefix.

3.4.3 Limitations of RPKI. RPKI only validates the origin and not the entire path. Therefore, it is still possible to impersonate the origin AS with all its prefixes in its entirety. If we review the previously mentioned BGP attacks, RPKI can only provide protection against the traditional sub-prefix hijack and traditional equally-specific-prefix hijack [15]. The other attacks will be considered as valid. Another type of traffic manipulation that RPKI filtering is unable to solve is the possibility to manipulate the path of traffic as achieved with the AS-path poisoning attack.

3.5 DNS, DNSSEC and RPKI

DNS is a vital element within the core Internet infrastructure. Almost all network services rely on DNS to translate domain names into IP addresses. Since most IP requests start with a DNS lookup, it is a very effective method to redirect traffic for nefarious reasons. A method of doing so is DNS cache poisoning (a.k.a. spoofing), where the traffic between the authoritative DNS server and the recursive resolver is being spoofed and injected with a false IP. This IP address then remains in the recursive resolvers cache further poisoning any DNS requests for that specific domain. To counter this, DNSSEC has had been developed as a method to verify domain records with cryptographic signatures. However, this will only work for domains which have DNSSEC signed.

The signing of DNSSEC happens at the level of individual domain names, and for this to work the top level domains also need to support DNSSEC. RPKI is enabled on the router level, and can protect multiple domains, if they reside within an RPKI protected AS, from prefix hijacks even when the domain is not DNSSEC signed.

4 RELATED WORK

Although there is, to the best of our knowledge, no earlier research conducted on measuring the status of RPKI filtering of DNS resolvers specifically, there are a few studies which go into detail on how they measured RPKI filtering and are therefore considered relevant to this research.

Multiple studies have tried various approaches to measure the adoption of RPKI filtering. The first study we will discuss was done by Gilad et al. [22]. The authors reviewed publicly available BGP path advertisements, comparing valid and invalid routes. They then identify another AS that appears on the path towards the valid prefix, but not towards the invalid prefix. The assumption is that these ASes perform filtering on these invalid routes. However, the limitation of this research is that filtering is not the only explanation that these invalid routes are missing, another explanation could be traffic engineering. Their methodology lacks proper control of the routes analyzed.

Reuter et al. [1] also analyzed the adoption of RPKI filtering, but recognized the previously mentioned flaw and improved on this by announcing their own valid and invalid prefixes. The advantage of this approach is that they were in control of the prefix validity, allowing them to expose filtering policies. By being in control of the prefix they could alternate the state, where the invalid prefixes could be alternated as also being valid and keeping a valid prefix for reference. Any changes in routes must therefore have been caused by RPKI filtering.

To further improve on these measurements Hlavacek et al. [23] increased the number of vantage points compared to the research of Reuter et al. This was accomplished with RIPE Atlas probes performing a traceroute to both a valid and invalid prefix. Another method they applied to increase the number of vantage points, was by sending TCP initiation segments with the SYN flag set to a selection of the top 1.25M Alexa websites from the valid and invalid prefixes. The TCP SYN and ACK replies are then captured at the source address within the aforementioned prefixes. By performing these real routing measurements the amount of vantage points to analyze traffic from could be increased, whereas previous research relied on limited vantage points as derived from RouteViews [24].

5 METHODOLOGY

In this section we will discuss the methodology used in our research. First, we will explain our environment (test setup). Second, we will describe the RIPE Atlas probe environment. We will conclude this section with a description of the conducted experiment.

5.1 Environment

To determine which DNS resolvers were protected by RPKI filtering, we designed a controlled experiment. This experiment consisted of two main elements, namely a server running as an authoritative DNS and BGP routing server (from now on referred to as the beacon), and RIPE Atlas probes.

The first element, the beacon, announced two RPKI signed prefixes. One of the announced prefixes is covered by a valid ROA

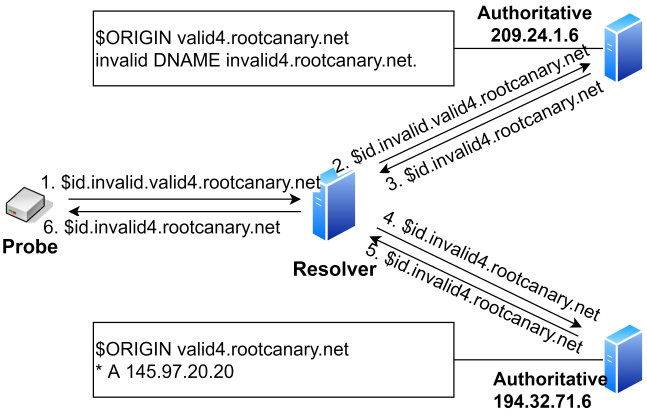

Figure 1: DNS query flow

whereas the other prefix is covered by an invalid ROA. The beacon is authoritative for two DNS domains: valid4.rootcanary.net hosted on the valid prefix and invalid4.rootcanary.net hosted on the invalid prefix. The beacon is located in the United States. The beacon has one upstream provider which does not filter, being NTT (AS2914). The valid prefix address being reached is 209.24.1.6 and the invalid is 145.97.20.20 as seen in Figure 1.

The second element, RIPE Atlas probes, which are devices used for Internet measurements, will first resolve an A record, named $id.invalid.valid4.rootcanary.net, in the valid prefix. The answer of this query will be a CNAME to a Resource Record (RR) in the invalid prefix. The CNAME is synthesized from a DNAME record which facilitates using unique query names from the RIPE Atlas probes, from which the unique (random id) part, by CNAME synthesis, will be carried to the domain at the invalid prefix. This enables associating the resolvers seen at both the valid and invalid prefixes, which in turn simplifies determination of whether it is subject to RPKI filtering or not. Furthermore, unique query names limit the possibility of caching. To further minimize caching, a time to live (TTL) of one second was configured for the RRs.

In our setup, we measured the queries sent to the on the RIPE Atlas probe's configured resolvers, and the queries reaching the authoritative server on the beacon. Everything in between is unknown. Figure 2 depicts an example of how a query might possibly be sent from the probe and eventually reach the beacon. It is not the goal of this research to gain an understanding of this architecture. We only need to establish whether the query that has been sent to one of the probe's resolver, reaches the invalid domain on the beacon. If it does, then that resolver of that probe is not protected by RPKI filtering.

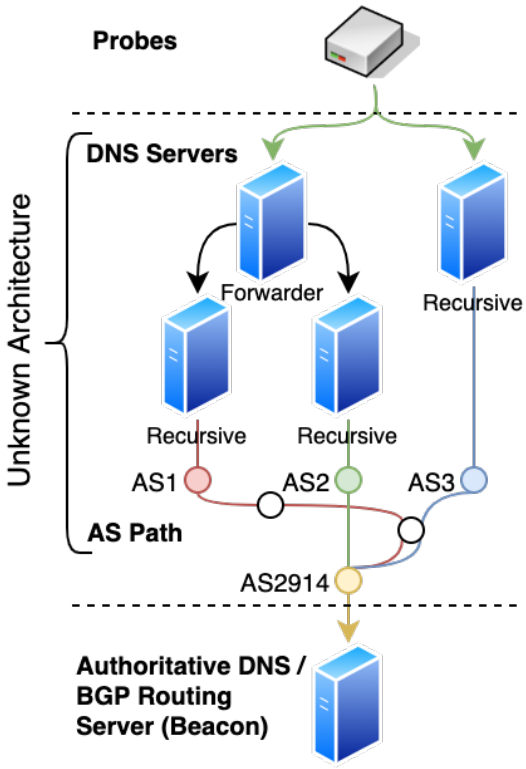

Figure 2: Test setup

5.2 RIPE Atlas Probe Population

The test setup consisted out of the entire set of RIPE Atlas probes with an IPv4 connection, totaling approximately 11000 individual probes as of February 3rd. Each probe sends a query every hour to its respective resolvers. The population varies from measurement to measurement. This will either stagnate or grow in line with the total RIPE Atlas probe population. The Atlas probes are represented for 50% by the following countries: Germany (14%), United States (12%), France (8%), United Kingdom (6%), Netherlands (5%) and Russia (5%). This indicates that the Atlas probe has limited representation in Africa, Asia and Middle and South-America and is mainly biased towards Europe and North-America.

5.3 Experiment

This subsection discusses how the experiment was conducted. First, we will discuss how the data was gathered. Second, we will explain how we analyzed the data.

5.3.1 Data Collection. The RIPE Atlas probes were instructed to query $id.invalid.valid4.rootcanary.net. The $id was constructed in the following manner: <random hexadecimal>-<timestamp>-<probe ID>. With this construction, the queries were unique and could be filtered based on time and probe ID. The query then reaches the beacon, for rootcanary.net, via the resolver configured on the probe. The beacon continuously made a tcpdump of all incoming DNS traffic from the recursive IP resolvers and rotates the captures hourly in accordance with the RIPE Atlas probe timing. The beacon also dumped the BGP table for every hour of pcap captures. Therefore, it was possible to determine the BGP path and length to the recursive resolver from the perspective of the beacon.

5.3.2 Data Parsing/Analysis. After collecting the pcaps the data was parsed, to export the queries to the valid and possibly invalid domain. This also contained the query ID and recursive IP from which it originated. The BGP table was also parsed to associate the resolver IP addresses with the prefixes as found in the BGP table dump. This gives an indication as to how the traffic might flow, including both the path and path length, from the recursive resolvers to the beacon. This is merely an indication since the flow of Internet traffic is asymmetric [25]. However, it is a reasonable metric to determine the path, since it is not possible to obtain the routes from each individual recursive DNS server to the beacon.

From the gathered data, two distinct view points were defined. One is from the perspective of the probe. From this perspective the probe is either not, partially or fully protected. To further clarify this, the following three scenarios will be explained. Suppose that the probes have two DNS resolvers configured. Scenario one, if both of the DNS queries sent by the probes can reach the invalid, the probe is not protected. Scenario two, if one of the DNS queries cannot reach the invalid but the second can, the probe is partially protected. Effectively, the probe itself is not protected since it can reach the invalid but it is categorized as partially protected. Scenario

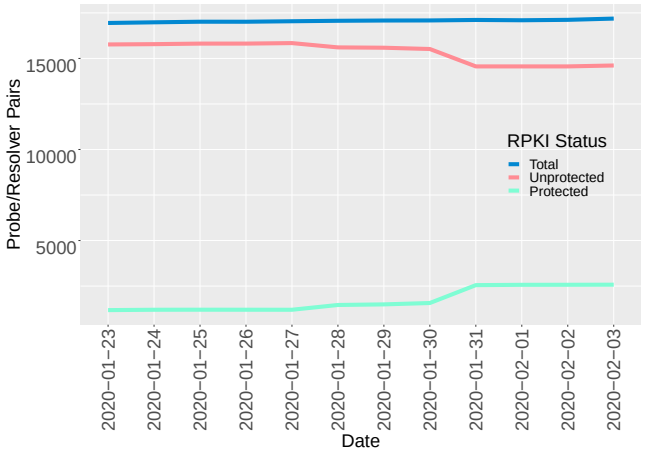

Figure 3: Probe/resolver pairs

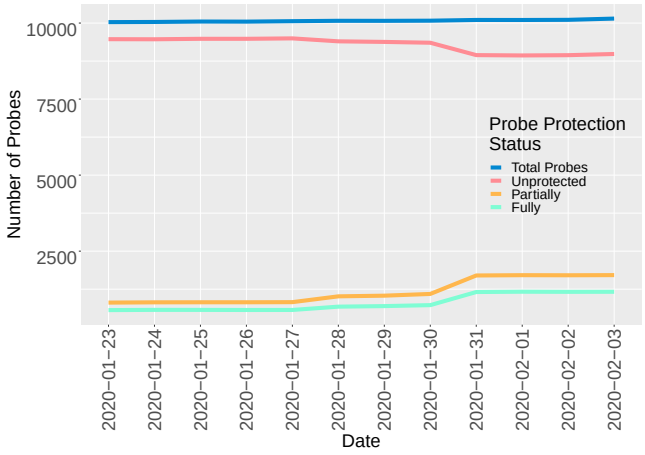

Figure 4: Probe time series

three, if none of the DNS queries sent by the probe can reach the invalid then therefore the probe is fully protected.

The second view point is from the perspective of the beacon. This viewpoint contained all received queries except for duplicate queries that where both received on the valid and invalid prefixes. In this case the valid queries were omitted, since their uniqueness was lost, therefore we could not correlate them.

Both these view points were analyzed with Python scripts, which calculated the 24 hour mean of each day. These scripts can be found on the following GitHub page [26].

6 RESULTS

This section illustrates the results from our experiment. All graphs are based on data from the 3rd of February 2020 with exception of the time series graphs, which contains data from the 23rd of January till the 3rd of February.

6.1 General RPKI DNS Resolver Coverage

Figure 3 visualizes the data in terms of total, protected and unprotected resolvers as configured on the probes. The figure shows that on the 23rd of January, 1179 (7%) of the 16947 probe configured resolvers were protected by RPKI filtering. On the 3rd of February, 2575 (15%) of the 17188 probes were protected.

Figure 4 visualizes the data in terms of total, fully, partially and unprotected probe configured resolvers. These results were generated by 10147 probes. Of these probes, 1712 had at least one protected resolver and 8983 had at least one unprotected resolver. This results in probes having both a protected and unprotected resolver. The fully protected are a subset of the partially protected, totalling to 1164 (11.5%) probes which were fully protected.

Figure 5 illustrates the top ten most popular ASes in terms of query amount. These ASes were responsible for 13185 queries, which is 42% of the total amount of queries. Google (AS15169) handles the most significant portion of queries, namely 17%. Next, Cloudflare (AS13335) handles 13.5% and OpenDNS (AS36692) handles 2.5%. Of the top ten ASes, Proxad (AS12322) protects the most queries in terms of ratio.

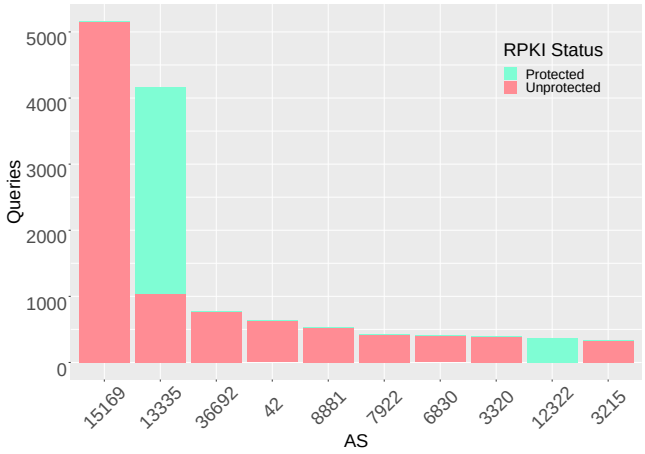

Figure 5: Top ten most popular ASes

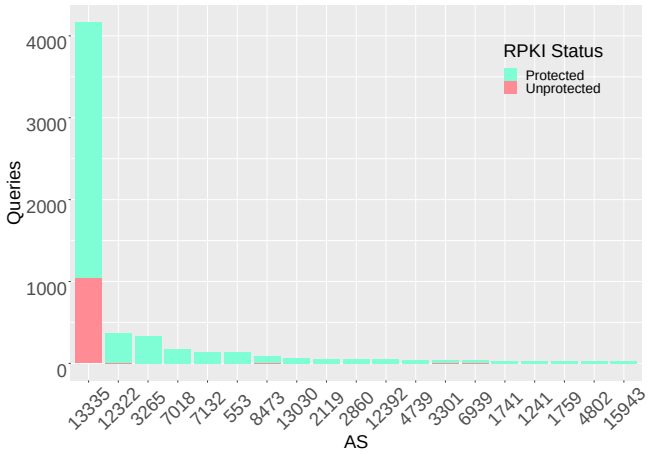

Figure 6: Top ten most protected ASes

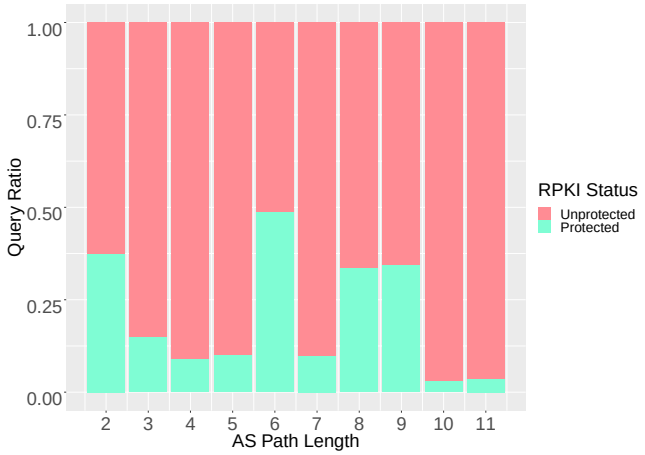

Figure 7: Relationship RPKI protection and AS path length

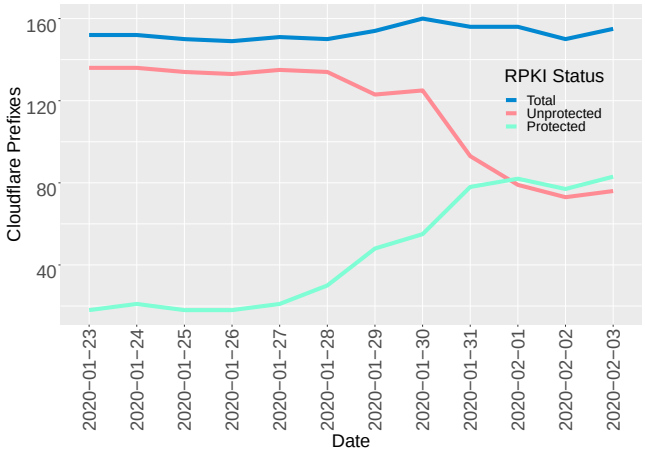

Figure 8: Cloudflare resolver prefix time series

Figure 6 illustrates the most protected ASes in terms of protection ratio, with a minimum protection ratio of 70%. Only two of the top ten most popular DNS resolvers ASes can be found in the top nineteen most protected ASes in terms of ratio, namely Cloudflare and Proxad (AS12322). It should be noted that the ASes in the figure required a minimum of 25 queries, which was our cut-off point. Otherwise, some ASes with a minimal amount of queries, e.g. one, would be in this figure. All other ASes with a minimum amount of 25 queries had less than 10% RPKI filtering ratio. All but one (Cloudflare) of the ASes in the figure provide full RPKI protection.

6.2 Influence of AS Path Length

Figure 7 illustrates the relationship between AS path length and whether the query was protected or not. It should be noted that queries that traverse a path length of two, three or four occur far more than the other path lengths. Namely, path length two occurred 6855 times, path length three 15828 times and path length four 4934 times. In contrast, path length five occurs 1462 times, path length six 891 times, path length seven 266 times and path length eighth 306 times. The remaining path lengths had less than fifty queries.

6.3 Influence of Anycast

To investigate the influence of anycast on RPKI filtering, we chose to focus on the DNS resolvers of Cloudflare, since they claim to have implemented RPKI filtering in their network [27]. With our measurements, we have received queries from 3425 different resolver back-end IP addresses, which can be segmented in-between 149 and 160 unique different IP prefixes depending on the day of measurement. We made the assumption that every IP prefix is an anycast Point of Presence (POP). On January the 23rd, 18 out of the 152 prefixes were protected by RPKI filtering as depicted in Figure 8. On February the 3rd, 83 out of the 154 prefixes were protected by RPKI filtering. This resulted in an RPKI protection coverage of 54% of their prefixes.

7 DISCUSSION

In this paper we researched the state of RPKI protection on DNS resolvers. Our results show that the RPKI coverage of the total amount of probes was 15%. However, this does not mean that 15% of the probes were protected. For example, when a probe has more than one resolver configured, it could be that one of its resolvers was protected and the other(s) not. This means that the probe would get an A record from the invalid IP address and therefore the probe is not protected. The actual percentage of fully protected probes is 11.5%. However, for redundancy reasons one could configure several DNS resolvers on their client, this may result in one resolver being protected and the other(s) not. Ultimately, if one of the resolvers is not protected, the client is not protected. Therefore, if possible, one should configure multiple RPKI protected DNS resolvers.

The top ten ASes, which handle 42% of the total queries, only support limited RPKI protection. In terms of amount of queries, Cloudflare provides the best protection. In terms of ratio, multiple ASes were fully protected. However, only three ASes that were fully protected, had received more than 150 queries. Combined, the top 18 ASes in terms of ratio protect 1729 queries. This is not even half of the total amount of queries that Google handles. To gain significant growth in terms of RPKI protection on DNS resolvers, one of the top ten most popular ASes would need to implement RPKI filtering.

Our results show that on January the 23rd till January the 30th there was slight growth in terms of unique queries protected, increasing from 7% to 9%. However, from January the 31st and onwards there was a increase to 15%. We have seen that this growth can mainly be attributed to Cloudflare, which grew from January the 30th from 44% to 71% on February the 1st.

Beforehand, we hypothesized that if the the AS path length between the resolver to our beacon increases, there would be a more significant chance that a query would be filtered by a traversing network. In contrast to our hypothesis, our results show that there is no correlation. This could potentially be due to the fact that some ASes may have filtered the route to the invalid prefix, but some

non-filtered ASes still advertise the route, and therefore a detour AS path is taken to reach the prefix.

As can be seen in our results, the influence of anycast on RPKI protection on DNS resolvers is notable. Currently, there is no guarantee that when someone configures their client with a Cloudflare resolver, that they have protection. This depends on the location of the client. Our results support that the RPKI filtering coverage of Cloudflare is growing, and therefore the chance of hitting an RPKI protected resolver back-end increases.

In general, it is interesting to see that ASes are either protected or unprotected. One notable exception being Cloudflare. We had expected more partially protected ASes. One explanation for this could be, as mentioned earlier, that when ASes in-between provide RPKI filtering another non-filtered detour path will be taken. In such cases, only if an AS is completely surrounded by RPKI filtering ASes it may have an impact.

7.1 Limitations

This research had several limitations. First, while the RIPE Atlas probes provide a good representation of diversity, the probes are more prevalent in Europe and North-America, and thus not a representation of the entire Internet [28]. The probes are also placed by tech aware people, which could lead to geek bias.

Second, to determine the path length from the resolver to the beacon, we were limited to use the reverse path. While this gives a reasonable estimation of the AS path taken, the Internet is asymmetric, and therefore another path from the resolver to the beacon could have been taken [25].

Third, in the preliminary phase of our research, we saw some situations in which DNAME caching was used. To exclude the effect of this on our research, we set the DNAME cache time to one second. Still, we cannot exclude that some queries may be answered within this time frame. Meaning that queries would only be sent to the invalid and not the valid prefix.

8 CONCLUSION

The goal of this research was to determine the state of DNS resolvers which were protected by RPKI filtering. The motivation behind this question is based off real world attacks, such as the Amazon Route 53 hijack, which could have been prevented if RPKI filtering was enabled at the recursive DNS resolvers. To the best of our knowledge there is no public data available indicating which DNS resolvers are protected by RPKI filtering. With the use of RIPE Atlas probes we were able to validate if the preconfigured recursive DNS servers were protected by RPKI filtering.

To support our main research question we answered the following sub-question: "How does anycast influence the protection of DNS resolvers?". Cloudflare was chosen as a suitable candidate for this sub question due to its sudden uptake in RPKI filtering. We identified that on the 3rd of February 83 of the 152 visible Cloudflare prefixes could be attributed to RPKI filtering. Our results show that Cloudflare does not fully filter all their prefixes yet. Therefore, at the time of writing, not all clients using Cloudflare's DNS resolver have RPKI protection. It is dependent on their location and which anycast POP they hit. Due to the nature of anycast routing, opting to choose the least-expensive route, it is uncertain that the clients' queries will always hit the same back-end DNS resolvers. It is dependent on the location from which the query originates. Thus, our results show that choosing an anycast DNS resolver that does not filter all its prefixes will in turn make it unable to guarantee 100% protection.

To research this further we posed the following sub question "How does the length of the AS path between resolver and authoritative server influence the level of RPKI protection?". The results do not indicate a correlation of AS path length influencing the level of RPKI protection.

To sum up all of these answers, we can answer the main research question "What is the state of RPKI filtering on DNS resolvers?". The results on January 23rd indicated that 7% of probe DNS queries were protected. On February 3rd we measured that 15% of the probe queries were protected. This indicates that the state of RPKI filtering on DNS resolvers is still in the minority but capable of quick growth. We attributed this growth to Cloudflare's implementation of RPKI filtering. There is still much room for improvement as a majority of the largest DNS resolvers, in terms of queries handled, do not partake in RPKI filtering.

9 FUTURE WORK

This study gives an indication on what the state of RPKI protection on DNS resolvers is. The research could be further improved in several ways.

Firstly, we noticed a few networks that would hit the valid prefix from different IP addresses several times, until they eventually hit the invalid prefix with an unprotected IP address. In our current approach, such a probe configured resolver is rightfully marked as not protected, but only the IP address hitting the invalid was used. Thus, even though part of the resolver architecture did appear to be protected, we did not register it as so. Therefore, our research is conservative, and the actual percentage of RPKI protected queries might be higher. One could potentially solve this problem by focusing on the AS or IP prefix the query was originated, instead of IP address.

Secondly, this study makes use of the RIPE Atlas probe environment. As mentioned before, the probes are mainly located in Europe and North America. One could try another measurement more focused on the protection in other geographical locations.

Thirdly, due to time and resource constraints, we scoped ourselves to only research IPv4 and not IPv6. However, it would also be interesting to research the state of IPv6 in regard to RPKI filtering DNS resolvers. By leaving out IPv6 resolving, our results may have been skewed. An IPv4 resolver may be protected by RPKI filtering, while the IPv6 resolver may not. With our current methodology, such a resolver would be marked as not protected. Vice versa, there might be resolvers which are protected for IPv6, but not for IPv4. We recommend a more nuanced investigation as outlined in our first future research suggestion to identify such dynamics.

ACKNOWLEDGMENTS

This research was made possible by several individuals and organizations. First of all, we like to thank our supervisors Willem Toorop and Roland van Rijswijk of NLnet Labs for their, creativity, guidance and contributions to the Python scripts. Secondly, we would like to thank Job Snijders of NTT for his beacon which we could use as an authoritative name server and BGP router. Thirdly, we like to thank Emile Aben of RIPE NCC for his resources to instruct the RIPE Atlas probes to do a full measurement every hour.

REFERENCES

[1] Andreas Reuter, Randy Bush, Italo Cunha, Ethan Katz-Bassett, Thomas C Schmidt, and Matthias Wählisch. Towards a rigorous methodology for measuring adoption of RPKI route validation and filtering. *ACM SIGCOMM Computer Communication Review*, 48(1):19–27, 2018.

[2] George Chang, Majid Arianezhad, and Ljiljana Trajković. Using resource public key infrastructure for secure border gateway protocol. In *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6. IEEE, 2016.

[3] Pavlos Sermpezis, Vasileios Kotronis, Alberto Dainotti, and Xenofontas Dimitropoulos. A survey among network operators on BGP prefix hijacking. *ACM SIGCOMM Computer Communication Review*, 48(1):64–69, 2018.

[4] Job Snijders. CloudFlare issues?, 2019.

[5] M Lepinski and K Sriram. RFC 8205: BGPsec Protocol Specification, 2017.

[6] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, IETF, February 2012.

[7] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Let the market drive deployment: A strategy for transitioning to BGP security. *ACM SIGCOMM computer communication review*, 41(4):14–25, 2011.

[8] Robert Lychev, Sharon Goldberg, and Michael Schapira. BGP security in partial deployment: Is the juice worth the squeeze? In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 171–182. ACM, 2013.

[9] Masahito Ando, Masayuki Okada, and Akira Kanaoka. Simulation Study of BGP Origin Validation Effect Against Mis-Origination with Internet Topology. In *2017 12th Asia Joint Conference on Information Security (AsiaJCIS)*, pages 75–82. IEEE, 2017.

[10] Taejoong Chung, Emile Aben, Tim Bruijnzeels, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, Roland van Rijswijk-Deij, John Rula, et al. RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins. In *Proceedings of the Internet Measurement Conference*, pages 406–419, 2019.

[11] Cloudflare. BGP leaks and cryptocurrencies, 2018.

[12] Internet Society. What Happened? The Amazon Route 53 BGP Hijack to Take Over Ethereum Cryptocurrency Wallets, 2018.

[13] Daniele Iamartino, Cristel Pelsser, and Randy Bush. Measuring bgp route origin registration and validation. In *International Conference on Passive and Active Network Measurement*, pages 28–40. Springer, 2015.

[14] D. McPherson E. Osterweil K. Sriram, D. Montgomery and B. Dickson. Problem Definition and Classification of BGP Route Leaks. RFC 7908, IETF, June 2016.

[15] Henry Birge-Lee, Yixin Sun, Anne Edmundson, Jennifer Rexford, and Prateek Mittal. Bamboozling certificate authorities with {BGP}. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 833–849, 2018.

[16] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. BGP hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*, pages 25–32. IEEE, 2019.

[17] Cloudflare. IP Ranges, 2019.

[18] R White. Architecture and deployment considerations for secure origin bgp. *IETF Internet draft: draft-white-sobgp-architecture-01*, 2006.

[19] Charles Lynn, Joanne Mikkelson, and Karen Seo. Secure bgp (s-bgp). *IETF Draft: draft-clynn-s-bgp-protocol-01. txt*, 2003.

[20] W George and S Murphy. Bgpsec considerations for autonomous system (as) migration. Technical report, RFC 8206, DOI 10.17487/RFC8206, September 2017, https://www.rfc-editor.org/info/rfc8206, 2017.

[21] L. Joncheray J-M. Jouanigot D. Karrenberg M. Terpstra T. Bates, E. Gerich and J. Yu. Representation of IP Routing Policies in a Routing Registry (ripe-81++). RFC 1786, Network Working Group, March 1995.

[22] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are We There Yet? On RPKI's Deployment and Security. In *NDSS*, 2017.

[23] Tomas Hlavacek, Amir Herzberg, Haya Shulman, and Michael Waidner. Practical experience: Methodologies for measuring route origin validation. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 634–641. IEEE, 2018.

[24] University of Oregon. University of Oregon Route Views Project, 2020.

[25] Wouter Bastiaan de Vries. Improving anycast with measurements. 2019.

[26] NLnetLabs. NLnetLabs/rpki-dns-test, 2020.

[27] Cloudflare. RPKI - The required cryptographic upgrade to BGP routing, 2018.

[28] RIPE NCC. Percentage of connected probes per country, 2020.