Spoofing Detection at IXPs:
A Reproducibility Study

Jasper Eumann

jasper.eumann@haw-hamburg.de

Hamburg University of Applied Sciences
Faculty of Engineering and Computer Sciences
Department Computer Science

Abstract. IP spoofing is often used to perform reflection- and amplification-based distributed denial-of-service attack (DDoS) attacks. DDoS attacks are a big problem for the Internet infrastructure and difficult to prevent. Detecting and preventing address forgery is an approach to defending against spoofing-based attacks. Detecting IP packets with incorrect source addresses in the attacker networks is easy, but it is all the more difficult at Internet Exchange Points (IXPs) or in transit networks.
The paper *Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses* [1] by Lichtblau et al. introduces a new method to detect spoofed IP packets in inter-domain traffic. This work aims to reproduces their methodology and results. We did this under similar conditions as the authors at an IXP in Germany, but with different data sets and at different times.

Keywords: Spoofing detection, DDoS, IP-spoofing, reflection attack, amplification attack, inter-domain traffic, reproducibility study

1 Introduction

Distributed denial-of-service (DDoS) attacks are a major problem for the Internet infrastructure. Even large web services like *Github* are only conditionally able to defend themselves against large-volume DDoS attacks. On February 28, 2018, *Github* was unavailable for nearly 10 minutes. An amplification and reflection attack using memcached [2] generated 1.35 Tbps traffic carried in 126.9 million packets per second [3].

Amplification is a common mechanism to improve the strength of an attack. This type of attack exploits services or protocols and is often combined with *reflection attacks. Reflection attacks* use a regular service such as the Domain Name System (DNS) to send packets to the victim. This is achieved by setting the address of the victim as the source address (IP address spoofing). The service then directly responds to the victim. It is easy to detect spoofed requests in the source networks near the attacker as all locally allowed IP prefixes are known. However, it is very difficult to detect spoofed packets at Internet Exchange Points (IXPs) or in transit networks.

In the paper *Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses* [1] Franziska Lichtblau et al. presented a method to detect spoofed traffic on the Internet Core Network at the *Internet Measurement Conference (IMC)* 2017. In the context of this work we use the presented methodology to reproduce their results at a different IXP in Germany. They provided us a part of their analyse scripts, which we extended and tailored to our measurement point.

The rest of this paper is organized as follows. Section 2 gives an introduction to IP spoofing. Thereafter Section 3 introduces the method from Lichtblau et al. in detail and gives a short introduction to our data sets. Our results are presented in Section 4. We end this paper with a conclusion in Section 5.

2 IP Spoofing

The term IP spoofing describes a method of someone forging the source address of an IP packet. IP spoofing can be used to hide the sender address when the attacker directly sends packets to the victim or to perform reflection attacks via a service like DNS.

IP spoofing can only be performed with connection-less requests. The ability of IP spoofing to mask the sender address is often observed together with TCP SYN attacks [4]. In TCP SYN attacks the attacker sends many TCP SYN packets to the victim to establish a TCP connection, but never finish the handshake. The victim must maintain a lot of state, which eventually leads to a resource exhausting, thus making this attack very effective [5]. This attack uses TCP packets, but it is connection-less because the TCP session will never be established.

Most spoofing based attacks use UDP because it is connection-less and stateless. Services known for reflection-based amplification attacks with datagrams are NTP, Memcached, SNMP and DNS.

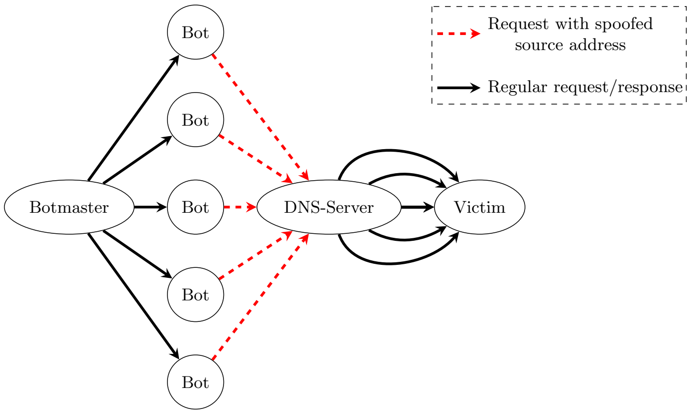


Fig. 1: Botnet-based amplification and reflection attack using a DNS server

Figure 1 shows a common reflection-based DDOS attack, using the DNS server for amplification. In this case, the attack is performed by a bot network controlled by the attacker (botmaster). When the botmaster initiates an attack the bots send spoofed packets to the DNS server (red dashed arrows). The DNS server responds to the victim with valid packets although the victim did not make the request. Only the packets to the reflection server are spoofed. Reflection turns a spoofed request into a valid looking response to the owner of the spoofed address.

2.1 Ingress and Egress Filtering

A way to detect and prevent IP spoofing is to identify the spoofed packets in the attacker's source networks and discard these illegitimate packets. A source network could be a stub network such as an Internet service provider (ISP) network. All globally available networks which make up the Internet are called autonomous systems (ASes). *The classic definition of an Autonomous System is a set of routers under a single technical administration using an interior gateway protocol (IGP) and common metrics to determine how to route packets within the AS and using an inter-AS routing protocol to determine how to route packets to other ASes* [6].

The main advantage of identifying attackers directly within their source network is that the attacks can be prevented directly and, if it is a distributed attack, it is mitigated at several points.

Two ways to accomplish this is to perform ingress and egress filtering. These methods are discussed below.

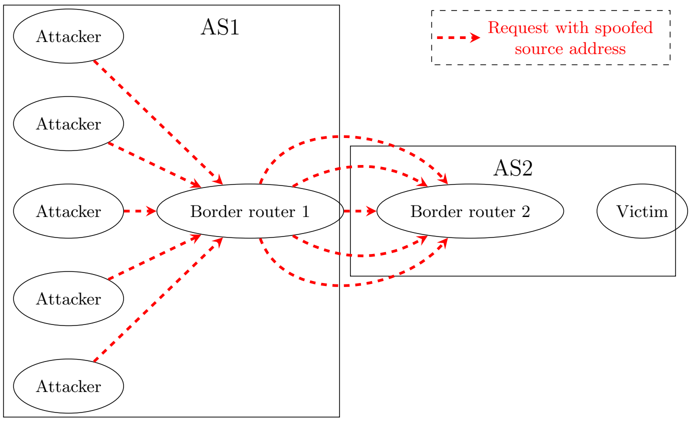


Fig. 2: A border router blocks incoming traffic using ingress filtering

Ingress filtering, as specified in RFC-2827 [7] and RFC-3704 [8] describes a method to detect incoming packets with unknown source addresses. Figure 2 shows how packets with spoofed addresses are discarded by ingress filtering from the border router of AS2, thus preventing the attack.

As a complement to ingress the egress method [9] prevents that traffic with spoofed address from leaving a network.

These methods are effective but not always deployed. 20% of the autonomous systems observed by Lichtblau et al. deploy ingress filtering and 50% have egress filtering active.

3 IP Spoofing detection at IXPs

Detecting spoofed packets in transport networks or at IXPs is a complex problem. An IP packet passes many routers and ASes on its way. The longer the distance from the source, the harder it is to check whether the source address is valid. All IP prefixes controlled by all ASes that my send legitimate traffic must be known to build ingress or egress filter rules.

Figure 3 shows traffic passing trough an IXP. The red dashes characterize a request with spoofed source address and the black arrows characterize the valid looking DNS server response. A spoofed packet can only be detected on the way from the attacker (in AS1) to the reflector (DNS server in AS3). The border router in AS4 cannot detect the malicious traffic because the packets are legitimately sent from the DNS server to the victim and cannot be distinguished from normal traffic. In this case, the ingress or egress filtering discussed in Section 2.1 is not possible for AS3, AS4 and at the IXP.

Lichtblau et al. [1] present a method to enable filtering in inter-domain traffic. This method uses the BGP path announcements as well as the produced AS relationship data of the method from Luckie et al. [10] described in Section 3.1, to build the customer cone for an AS.

The customer cone contains all legitimate IP prefixes of all ASes reachable via an AS. Incoming packets that have a source address not converted by a prefix in the customer cone of the previous AS can be considered as spoofed.

In the case of Figure 3, the customer cone of AS2 contains the announced IP prefixes of AS2 and AS1. Since the attacker uses source addresses of AS4, the IXP would be able to detect the

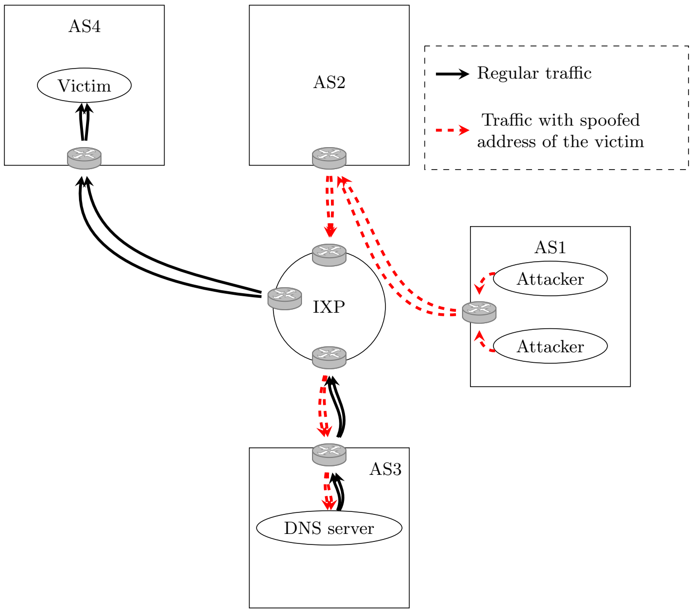


Fig. 3: A DNS amplification and reflection attack traversing an IXP

spoofed packets and could prevent the attack. The method and the data sets used are explained in detail in the following sections.

3.1 Data Sets

BGP: The Border Gateway protocol (BGP) is used to exchange information about IP prefixes and possible routes between ASes [6] on the Internet backbone. BGP data is usually used to create routing tables.

Lichtblau et al. use BGP path announcements and routing table dumps from different route monitors bundled by the BGPStream project [11]. This data contains IP prefixes, the AS number of proclaiming AS and the AS path to the origin AS of the IP prefix.

```
BGP4MP|1522454399|A|206.197.187.10|14061|185.160.179.0/24|14061 1299 12880 49148|IGP|206.197.187.10|0|0|||
```

Fig. 4: BGP path announcements in BGPDump [12] in one line representation format

Figure 4 shows a BGP path announcement for IP prefix *185.160.179.0/24*. With this announcement AS14061 propagates that it knows a path for addresses in the prefix via AS1299 and AS12880 to the owner AS49148. This forwarding information will be used to detect invalid traffic.

CAIDA Customer Cone: Luckie et al. [10] created a methodology to detect customer and peering relationships between ASes in their paper *AS Relationships, Customer Cones, and Validation*. They used five data sources to detect these relationships:

1. BGP path announcements

2. BGP community strings [13]
3. Routing Policy Specification Language (RPSL) [14]
4. Directly reported BGP peering and customer relationships
5. Publicly available RIPE WHOIS [15] information

Their methodology produces monthly updated publicly available data [16] and encloses private AS relationships.

Flow Samples: The network packets on our measurement point are captured and stored as sFlow samples [17]. The sFlow format contains the IP header of the IP packet without payload, but extended by data such as the current sample rate and the router IP address.

In our setup, sFlow samples are collected and stored on the hard disk in five minute intervals.

3.2 Detecting Spoofed Packets at IXPs

Lichtblau et al. use four categories to classify traffic:

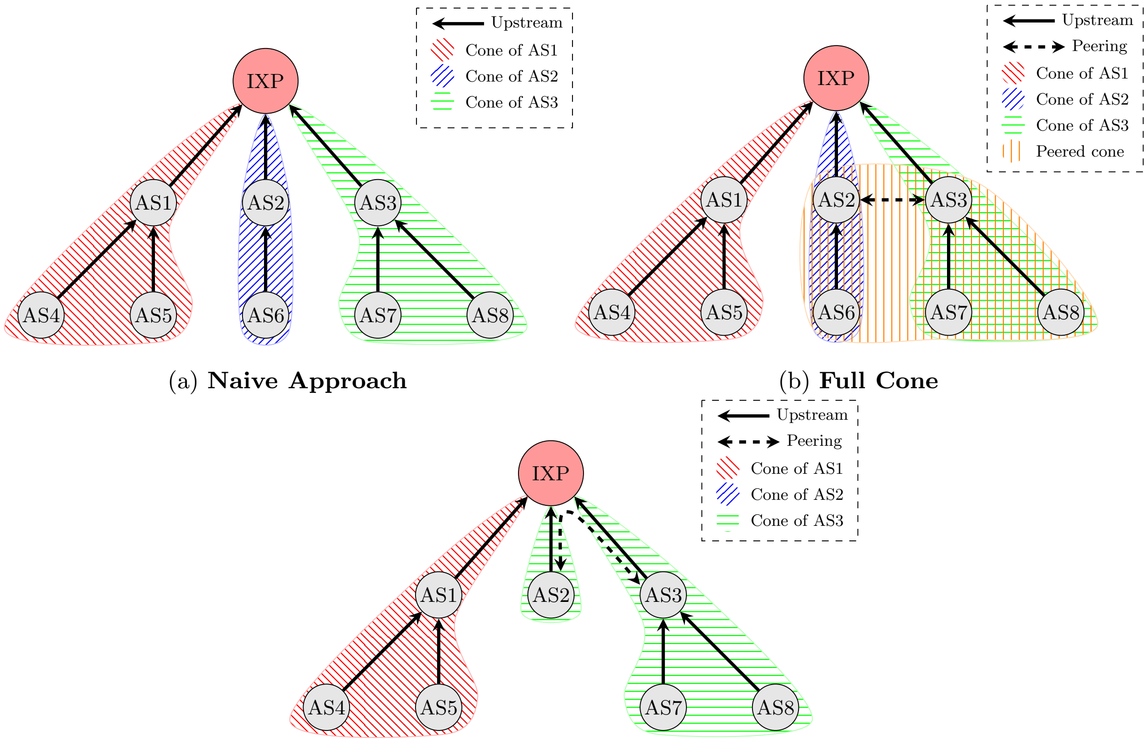
1. **Regular** contains legitimate traffic
2. **Invalid** denotes spoofed traffic
3. **Bogon** encloses traffic from private networks or other ineligible routable prefixes
4. **Unrouted** contains all traffic from unannounced prefixes

Their concept consist of three approaches and one extension. The approach names are taken from the paper.

1. **Naive Approach:** BGP path announcements are used to build a customer cone of all ASes behind the AS.
2. **CAIDA Customer Cone:** Uses the results of paper *AS Relationships, Customer Cones, and Validation* from Luckie et al. [10] to build the customer cone with a focus on customer-provider relationships.
3. **Full Cone:** Both previous approaches can produce false positives: the Naive Approach ignores asymmetric routes and the CAIDA Customer Cone might not cover all peering links. This approach adds bidirectional links between all neighbouring ASes and adds peering links contained in the CAIDA Customer Cone to fix these issues.
4. **Multi-AS Organization:** This extension enhanced the three previous approaches by taking companies with sibling ASes into account. This is achieved by adding connections between all ASes of a multi-AS organization and allowing bidirectional data exchange between them. The term multi-AS organization was determined in a paper from Cai et al. [18].

   The multi-AS organization information extension have no effect in combination with the full cone approach in our and in Lichtblau et al. results. The links of multi-AS organization ASes are already covered by the CAIDA customer data or the BGP data, so that the affected ASes are implicitly handled correctly.

A customer cone example from the perspective of an IXP is shown in Figure 5. The black arrows always point to the upstream of the current AS. An example for the **Naive Approach** is shown in Figure 5a. *AS4* and *AS5* sends traffic to *AS1* which accepts traffic for prefixes announced from *AS4* and *AS5*.



(a) **Naive Approach**

(b) **Full Cone**

(c) IXP member in the cone of another member

Fig. 5: Customer cone topology from the perspective of an IXP

In case of the **Full Cone** approach shown in Figure 5b, all traffic flows of Figure 5a remain possible. In addition, the peering relationships between the member ASes of the *IXP* are part of the cone. It is assumed that all peering links between ASes are bidirectional. This means that all traffic from *AS6* can legitimately arrive at *AS7* via *AS2* and the link between *AS3*.

The peering relationships are only important for checking the traffic between the involved ASes.

Traffic which reaches an AS via a peering link is usually not forwarded via the upstream AS of the AS since costs arise for the AS. Our measurements confirm this and while traffic is not normally directed upstream some AS still doing.

Figure 5c shows a possible edge case of a customer cone of an AS. AS2 is a customer of AS3 and AS2 and AS3 are connected via the IXP, possibly via a separate VLAN and with a private BGP peering session. This constellation leads to a possible misclassification of traffic and must be examined in our feature studies.

3.3 Execution and Structure of Our Measurements Study

Franziska Lichtblau and Thorben Krüger kindly provided us their scripts [19]. In order to reproduce their measurements, we used these scripts, which we extended by functionality that was needed for our specific measuring point or missing.

Their work allows construction of the customer cone from BGP data and includes an example. With the original scripts it is possible to validate an IP packet with the AS number of the AS via which the packet reached the IXP. The answer is either that the source IP of the packet is in the customer cone of the AS or not.

The customer cone build script does not distinguish between different peering relationships. Upstream and peering relationships are mapped in the same way. For this validation study, we have not changed this because we want to make as few adjustments as possible to obtain comparable results.

Following the descriptions in the paper we add functionality to:

1. perform bogon and unrouted filtering
2. fetch and convert CAIDA customer cone data
3. convert flow data
4. evaluate and display results

We use the structure shown in Figure 6 to perform our measurements.

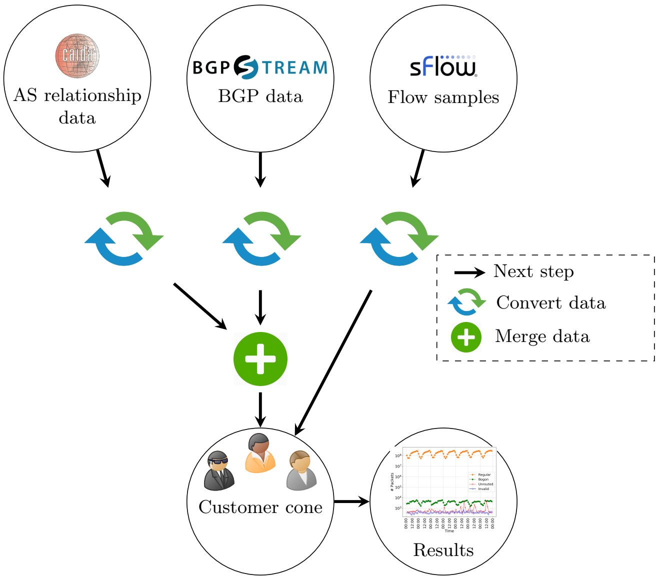


Fig. 6: Schematic representation of the measurements and validation

The upper edge of the figure shows the input data as described in Section 3.1. Flow samples contain the traffic headers that should be checked and must be converted to a regular text representation. The BGP and AS relationship data needs to be converted and merged to build the customer cone.

We only include the peering relationships of the IXP member ASes, because peering traffic is not routed to the upstream AS. As a result, peering relationships only affect ASes at the same level. These leads to an implementation inaccuracy that cannot be improved without a distinction between different peering relationships.

Each packet of converted sFlow data is checked against the customer cone formed from the combined data and the results are classified and written into a file. In this way, we can use these files to build our plots.

The customer cone must be rebuilt for each time interval. We use the same customer cone for a period of 24 hours and check all traffic in this interval against the same customer cone.

4 Comparative Results

We conduced our measurements at a medium-sized IXP in Germany. This contrasts Lichtblau et al. who performed their measurements at a large IXP also in Germany. The methodology should be transferable and not dependent on the size of the IXP as it should be able to detect spoofed traffic in any number of packets and affected ASes.

With only the necessary adaptation to our measurement environment, specifically sFlow data parsing and the mapping from interface to AS, we obtain the following distribution of invalid and regular traffic with the naive cone approach for February 2018 shown in Table 1.

Invalid	01.5%
Regular	98.5%

(a) Lichtblau et al.

Invalid	00.3354%
Regular	99.6646%

(b) Our results

Table 1: Distribution of invalid and regular traffic with the naive cone approach over a four-week period

Table 1 shows that we classified less traffic as invalid than Lichtblau et al. With the Naive Approach Lichtblau et al. found invalid traffic in 84% of all seen ASes. In comparison, we found spoofed traffic in 73% of all ASes of our IXP. Tab. 1 does not include bogon and unrouted traffic because the detection for these classes is not part of the original codebase. To make the results comparable, we add bogon and unrouted traffic from Lichtblau et al. to their invalid value.

With the full cone approach including bogon and unrouted traffic filtering, we receive the following results for February 2018 shown in Table 2.

Invalid	00.03%
Bogon	00.02%
Unrouted	00.02%
Regular	99.93%

(a) Lichtblau et al. results

Invalid	00.0001%
Bogon	00.0025%
Unrouted	00.0006%
Regular	99.9968%

(b) Our results with peering information

Invalid	00.0076%
Bogon	00.0025%
Unrouted	00.0006%
Regular	99.9893%

(c) Our results without peering information

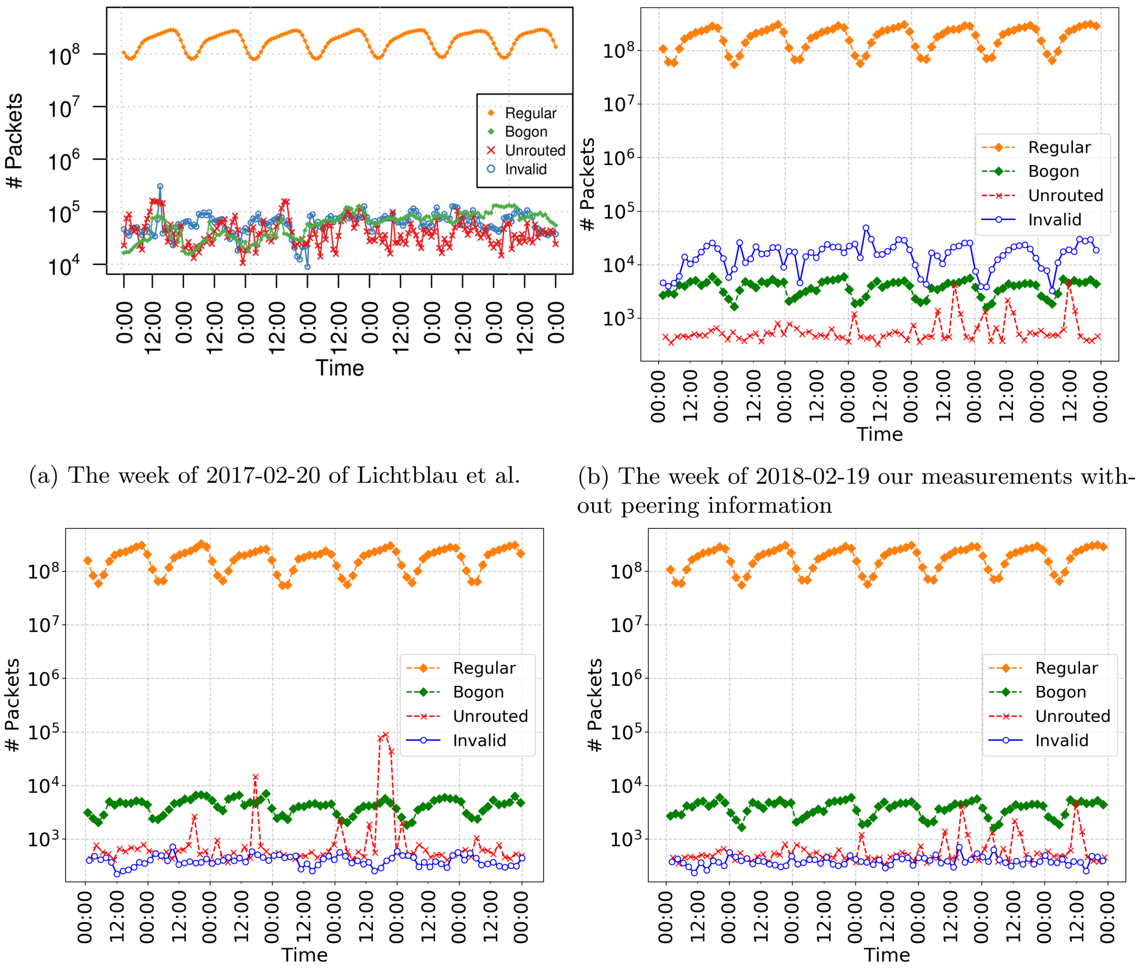
Table 2: Results with the full cone approach over a four-week period

Table 2 shows that we see two orders of magnitude less invalid and unrouted traffic than Lichtblau et al. Furthermore the bogon traffic is two orders of magnitude less with peering information.

In comparison, without peering information we see less invalid traffic than Lichtblau et al. but the values only differ by one order of magnitude.

We selected plots from the paper to reproduce as shown below. We cannot produce plots for the same time period because we do not have the traffic data for the specific times.

Figure 7 displays the packet count of all categories in sampled packets. The x-axis shows the time in 12 hour steps and the y-axis the number of packets. Our results are shown in Figure 7b and Figure 7d. We calculate the sample rate into our results because the sample server uses a flexible sample rate depending on the current load. To make our results comparable we scale our values with a static factor into the range of Lichtblau et al. results.



(a) The week of 2017-02-20 of Lichtblau et al.

(b) The week of 2018-02-19 our measurements without peering information

(c) Our results for the week of 2018-02-12 with peering information

(d) Week of 2018-02-19 our measurements with peering information

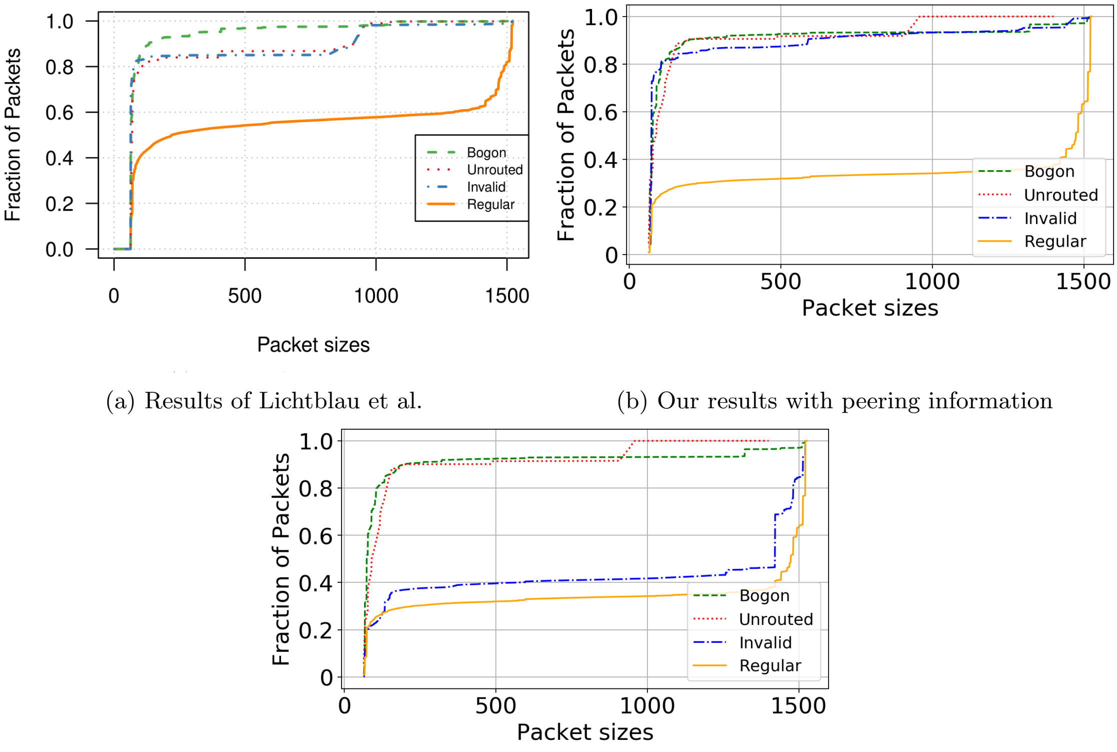
Fig. 7: Time series that shows the traffic class distribution over a week

At the top of all plots in Figure 7 is the regular traffic line. Bogon, invalid, and unrouted traffic are not so evenly distributed over the day.

Figure 7d shows the time series week of 2018-02-19 without peering information. We see less bogon and unrouted but a similar amount invalid traffic than Lichtblau et al. The invalid traffic is more than two orders of magnitude bigger than other results with peering information in Figure7b. This plot shows how big the impact of peering links is in our setup.

Figure 7b shows the same time series week of 2018-02-19 but with peering information. Unrouted and invalid traffic are one order of magnitude less than bogon traffic and peaks are visible. These characteristics are not stable. Figure 7c shows divergent traffic characteristics, with strong peaks with more than four orders of magnitude for the week time series of 2018-02-12 with peering information. The peaks stem from missing announcements in our BGP data. These variations in traffic characteristics are problematic and, in our opinion, depend on the completeness of the BGP data in use.

The fraction of packet sizes over a four-week period is shown as cumulative distribution function (CDF) plot in Figure 8. The x-axis denotes the packet size in bytes and the y-axis percentage distribution of that packet size in the current classification class. In comparison, our results with peering information show a similar distribution in regular traffic to Lichblau et al. but exhibit a lower curve. This means we have seen fewer small packets. The curves of bogon, invalid and unrouted traffic are similar. Overall the traffic for bogon, unrouted and invalid traffic mostly consist of small packets.



(a) Results of Lichtblau et al.

(b) Our results with peering information

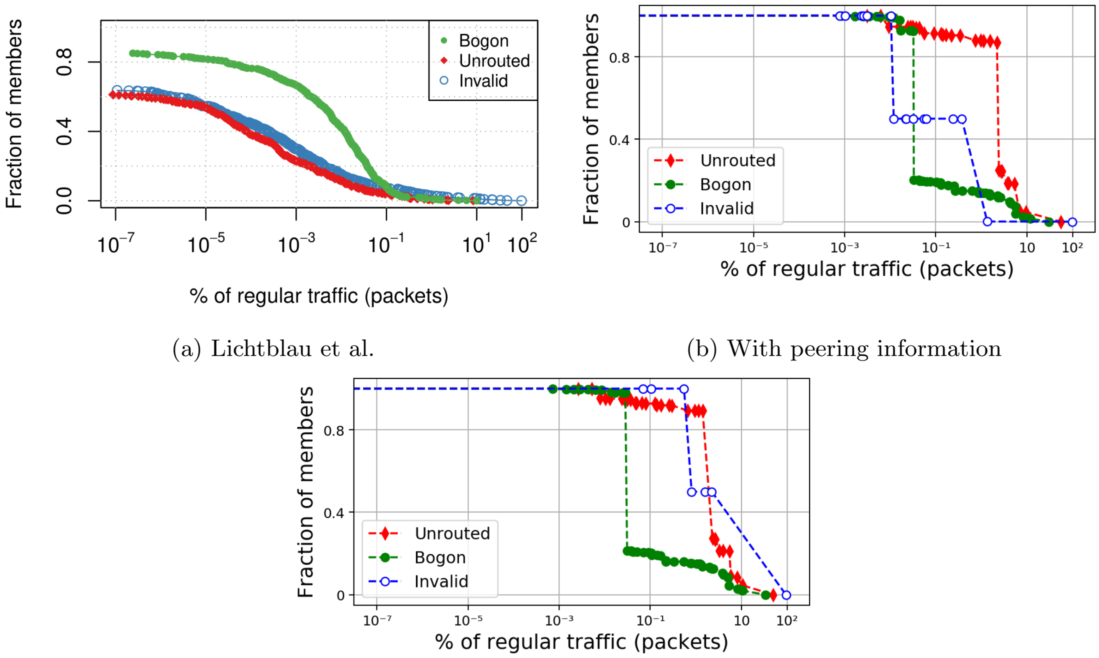
(c) Our results without peering information

Fig. 8: CDF: Fraction of packets by size and category

Figure 8c shows that the invalid traffic has a similar distribution to the regular curve for measurement without peering information. In our opinion this is an indication of a wrong traffic classification because the invalid traffic packet sizes are too similar to the regular traffic packet sizes. Opposed to Figure 8c, our results for these plots with peering information Figure 8b are similar to Lichtblau et al. Our curves almost match those of Lichtblau et al.

Figure 9 shows the fraction of IXP members that have at least in precent of their traffic classified as invalid, bogon and unrouted plotted as a complementary cumulative distribution function (CCDF). The x-axis shows the number of packets as a percentage of the regular traffic of an AS and the y-axis shows the percentage of ASes that have this ratio.

In the results of Lichtblau et al., a few ASes produce a large part of invalid traffic and a large percentage of IXP members in the bogon and unrouted categories come from few ASes. In our results this fraction is not as big but it is visible that most of the invalid traffic is produced by less than 20% of ASes. We have seen invalid traffic in much less ASes than Lichtblau et al. The peering information has little influence on the distribution. Our results with and without peering information in Figure 9b and Figure 9c look similar.



(a) Lichtblau et al.

(b) With peering information

(c) Without peering information

Fig. 9: CCDF: Fraction of bogon, unrouted and invalid of total traffic per IXP member AS

The following figures show traffic distributions grouped by TCP and UDP separated by source (SRC) and destination (DST) port. Figure 10 shows the graph from Lichtblau et al. while Figure 11 presents our results.

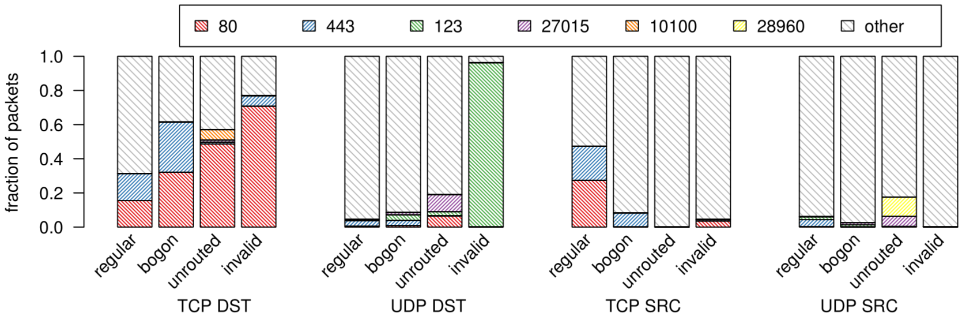
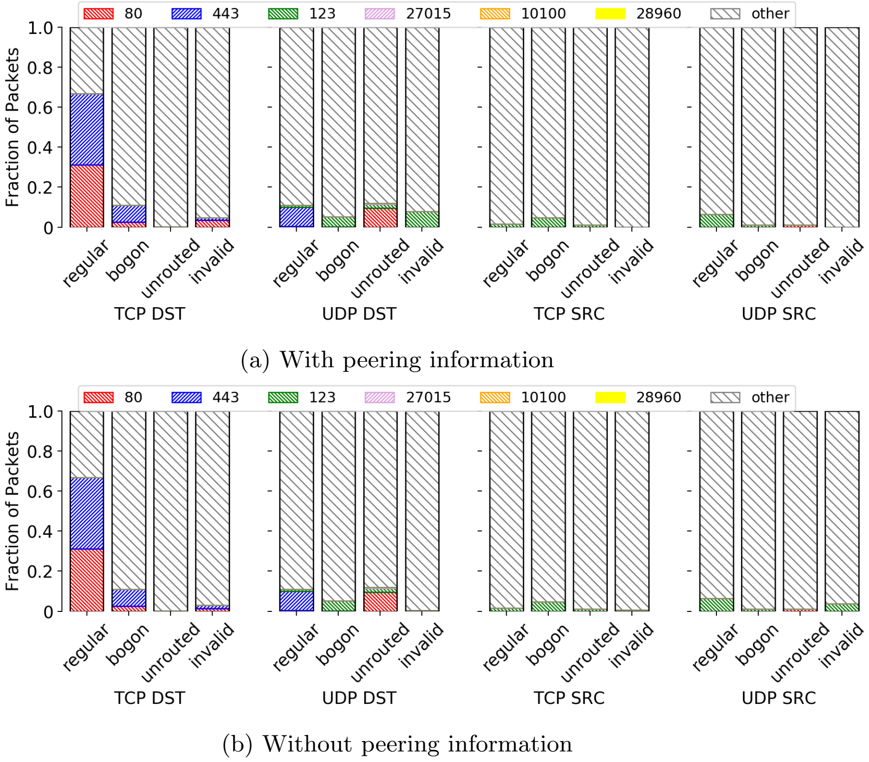


Fig. 10: Traffic mix for regular, bogon, unrouted and invalid traffic (Lichtblau et al.)

In Figure 10 and Figure 11 we see significantly different results. One possible reason is that we use a different measurement point. Additionally, the usage of protocols and services to perform reflection attacks has changed over one year. The Link11 DDoS report [20] shows that Memcached, Simple Service Discovery Protocol (SSDP) and Connection-less Lightweight Directory Access Protocol (CLDAP) are increasingly used to perform large-gain reflection-based DDoS attacks.

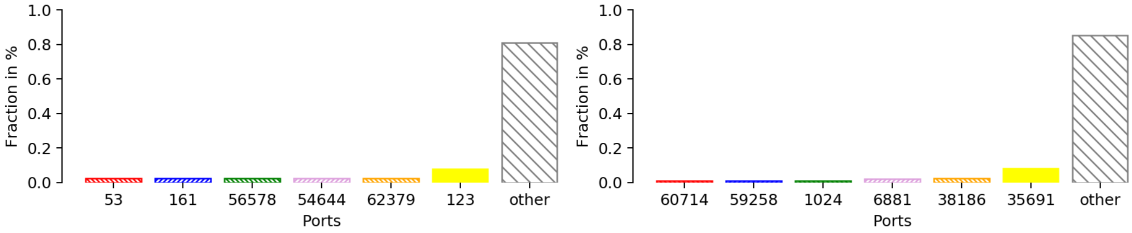


(a) With peering information

(b) Without peering information

Fig. 11: Traffic mix for regular, bogon, unrouted and invalid traffic (our results)

In our results most of the invalid traffic goes to ports in the dynamic or private ports range (49152–65535). So we cannot clearly assign the ports to a service. Without peering information, we do not see more than a few packets invalid UDP traffic to any of the selected ports. This is a possible indication of an incorrect classification because no known services for reflection-based amplification attacks are identifiable in our plot.

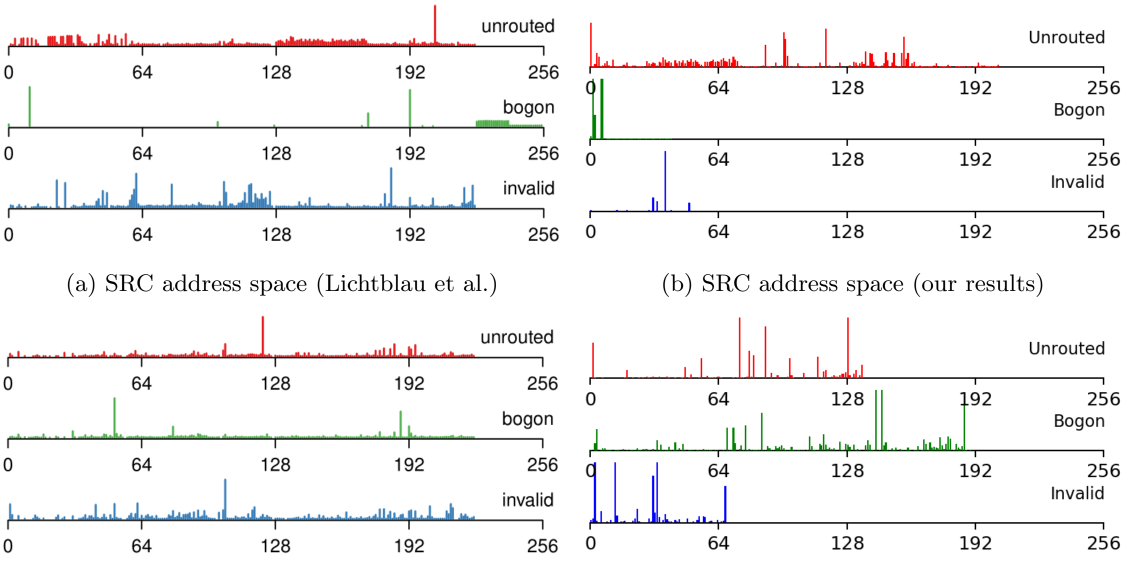


(a) With peering information

(b) Without peering information

Fig. 12: Traffic mix for our most frequently seen invalid UDP destination ports

Figure 12 shows the traffic mix for our most frequently observed UDP destination ports among the as invalid classified traffic, with and without peering information. In both cases there are large fluctuations in port utilization. With peering information we observe a large fraction of NTP traffic as well as some DNS and SNMP packets. Without peering information, only reserved traffic is identifiable.



(a) SRC address space (Lichtblau et al.)

(b) SRC address space (our results)

(c) DST address space (Lichtblau et al.)

(d) DST address space (our results)

Fig. 13: Fraction of source and destination IP prefixes

For these plots the address space is divided into /8 prefixes. Figure 13 shows the number of packets per prefix for source (a, b) and destination (c, d) addresses split into bogon, unrouted, and invalid traffic. The fraction for unrouted traffic is scattered across the IP space. In contrast, the bogon traffic distribution has few prefixes from which most of the the traffic addresses originates. The invalid SRC addresses originate from many prefixes in the measurements of Lichtblau et al. while we only see spoofed packets originate in a few prefixes.

In case of the DST address prefixes shown in Figure 13c and Figure 13d both plots show bogon traffic to be strongly scattered with a few peaks. The DST prefixes fraction of unrouted traffic in our results is mostly scattered between prefixe 0.0.0.0/8 and 128.0.0.0/8. In contrast, Lichtblau et al. see unrouted traffic from most prefixes.

In summary, we can say that the scattering over all prefixes as well as DST and SRC prefixes is lower in our results. This may be due to the fact that fewer different AS prefixes are routed or have other reasons at our measuring point or classification.

5 Conclusion

We obtain partly comparable results to Lichblau et al. Upon closer inspection, most of the packets classified as invalid may be incorrectly categorized. The correct use of peering information is unclear, but we get comparable results to Lichtblau et al. with and without peering information.

At first glance the methodology is able to detect spoofed traffic in the core Internet. Many of the packets classified as invalid originated from a few ASes for which the corresponding BGP announcement or peering information were missing. It is possible and probable that for invalid traffic we get a lot of false positives. The missing distinction between different peering relationships and the vague description of the use of CAIDA customer cone data makes implementation inaccuracies likely.

In our opinion the concept is good to get an overview over the current traffic mix at an IXP and a good guide to detect spoofed traffic. However, at its current state of implementation, it is not a safe classification method. The strong dependence on the completeness of the BPG data and the only in monthly intervals available Customer Cone data of CAIDA makes the concept error-prone. In our opinion it is necessary to consider peering and upstream relations when building the customer cone.

References

[1] F. Lichtblau, F. Streibelt, T. Krüger, P. Richter, and A. Feldmann, “Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses”, in *Proceedings of the 2017 Internet Measurement Conference*, ser. IMC ’17, London, United Kingdom: ACM, 2017, pp. 86–99.

[2] Memcached.org, *Memcached: Free and open source, high-performance, distributed memory object caching system*, Accessed: 2019-01-08, 2018. [Online]. Available: https://memcached.org/.

[3] S. Kottler, *GitHub Engineering February 28th DDoS Incident Report*, Accessed: 2019-01-08, 2018. [Online]. Available: https://githubengineering.com/ddos-incident-report/.

[4] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks.”, *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[5] J. Mirkovic and P. Reiher, “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms”, *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[6] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", IETF, RFC 4271, Jan. 2006.

[7] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", IETF, RFC 2827, May 2000.

[8] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks", IETF, RFC 3704, Mar. 2004.

[9] D. Distler, *Performing Egress Filtering*, Accessed: 2019-01-08, 2008. [Online]. Available: https://www.sans.org/reading-room/whitepapers/firewalls/performing-egress-filtering-32878.

[10] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k. claffy kc, "AS Relationships, Customer Cones, and Validation", in *Conference on Internet Measurement Conference*, ser. IMC'13, Barcelona, Spain: ACM, 2013, pp. 243–256.

[11] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti, "Bgpstream: A software framework for live and historical bgp data analysis", in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16, Santa Monica, California, USA: ACM, 2016, pp. 429–444, ISBN: 978-1-4503-4526-2.

[12] Y. Ohara, *Arbor networks - the security division of netscout*, Accessed: 2019-01-08, 2015. [Online]. Available: https://labs.ripe.net/Members/yasuhiro_ohara/bgpdump2.

[13] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute", IETF, RFC 1997, Aug. 1996.

[14] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, "Routing Policy Specification Language (RPSL)", IETF, RFC 2622, Jun. 1999.

[15] RIPE.net, *Ripe database query site*, Accessed: 2019-01-08, 2018. [Online]. Available: https://apps.db.ripe.net/db-web-ui/#/query.

[16] Caida.org, *Caida as relationships dataset*, Accessed: 2019-01-08, 2018. [Online]. Available: http://www.caida.org/data/as-relationships/.

[17] M. L. Peter Phaal, "sFlow Version 5", InMon Corp., Tech. Rep., Jul. 2004.

[18] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger, "Towards an AS-to-Organization Map", in *Proc. of the 10th ACM IMC*, New York, NY, USA: ACM, 2010, pp. 199–205.

[19] F. Lichtblau, F. Streibelt, T. Krüger, P. Richter, and A. Feldmann, *Transitive closure cone*, Accessed: 2019-01-08, 2018. [Online]. Available: https://gitlab.inet.tu-berlin.de/thorben/transitive_closure_cone.

[20] Link11.com, *Link11 ddos report for europe*, Accessed: 2019-01-08, 2018. [Online]. Available: https://www.link11.com/en/ddos-report/.