Network Visibility-aware Blacklist Generation

Pierre Edouard Fabre*†, Jouni Viinikka*, Hervé Debar† and Gregory Blanc†
*6cure
g4200 Herouville-Saint-Clair, France
Email: {pef, jvi}@6cure.com
† Institut Mines-Telecom, Telecom SudParis
CNRS Samovar UMR 5157, 91011 Evry , France
Email: {herve.debar,gregory.blanc}@telecom-sudparis.eu

***Abstract*—Volumetric Distributed Denial of Service (DDoS) attacks have become a major concern for network operators, as they endanger the network stability by causing severe congestion. Access Control Lists (ACLs), and especially blacklists, have been widely studied as a way of distributing filtering mechanisms at network entry points to alleviate the effect of DDoS attacks. Different blacklist generation approaches, as proposed in the literature, are dependent on the information available on the network traffic. Nonetheless, the collection of traffic information comes at a cost that increases with the level of detail. To study the impact of the level of detail available, we formulate three scenarios. Each scenario describes a typical collection granularity used by operators. We then define blacklist generation algorithms corresponding to each granularity. Scenarios are evaluated with a mix of real legitimate and generated attack traffic. The evaluation shows that the amount of information does have an impact on the attack filtering results, and that one should choose the blacklist generation algorithms in regard of the available level of detail. Experiments also show that having more information does not always translate to more efficient filtering.**

***Keywords*—volumetric DDoS; network monitoring; ACLs; blacklists.**

I. Introduction

The volume of bandwidth-depleting Distributed Denial of Service (DDoS) attacks has repeatedly reached new records in the recent years. In addition to disrupting the targeted service, these attacks can cause congestion at different points upstream from the actual target, creating wider perturbations.

Any mitigation solution downstream from a choke point will be ineffective [1], [2], as the saturation of an upstream link causes losses of legitimate traffic as well before it reaches the mitigation solution.

A distributed deployment of mitigation solutions could allow them to act before the funneling effect of attack traffic converging towards the target becomes too important. Although researchers have proposed distributed deployment strategies [3]–[5], the financial cost associated with the large number of nodes to deploy is often prohibitive.

Another option would be to use existing, widely deployed equipment, e.g., routers, for mitigation. Routers, for example, implement different mechanisms, such as FlowSpec and Access Control List (*ACL*), that can be used to drop potentially a large part of a volumetric DDoS attack, depending on the attack characteristics and thus to alleviate the congestion. The remaining part of the attack traffic may then be filtered with a more precise, dedicated solution [6]. The coarse granularity of filtering mechanisms available in network equipment is likely to cause collateral damage, i.e., legitimate traffic being filtered. Researchers have already worked on blacklist generation algorithms to create efficient filtering lists aiming at reducing collateral damage while maximizing the attack traffic filtering [7]. These algorithms typically take as input information on legitimate and malicious traffic, including lists of legitimate client and attacker IPs.

In this paper, we focus on the impact of *network visibility* on the blacklist generation problem. By network visibility we mean the availability of traffic information, and to the best of our knowledge, this impact has not been studied yet. We define and examine several scenarios reflecting the different levels of information a network operator has access to. We study the efficiency of blacklists deployed on a single node - distributed and/or collaborative filtering schemes are not considered. Finally, we provide means to find a trade-off between the level of visibility - increased visibility comes with increased cost - and efficiency of filtering.

The rest of the paper is organized as follows, Section II provides definition which our work is based on and the generic blacklist assumption. The Section III is an overview of the literature in the traffic filtering area. Section IV lays the fundamental problem of information availability to generate blacklists and formulate scenarios depicting levels of network visibility. In Section V, we detail blacklist generation scheme designed to fit in these scenarios. Section VI describes experiments and discusses their results. Finally, Section VII concludes the paper.

II. Background

Our study is focused on the mitigation of bandwidth-depleting DDoS attacks. This section provides definitions used in the remainder of this paper, describes the threat landscape, and states our underlying the assumptions.

A. Definitions

We will be using the following definitions in this paper.

Aggregate is a network address prefix aggregate as used in route aggregation.

Detection system is any system capable detecting volumetric DDoS attack and reporting the source and destination addresses participating in the attack. A detection system can be external to the network being monitored.

Monitoring system is any system providing network telemetry for the monitored system. For our needs, we expect the telemetry to include at least traffic volumes between source and destination addresses, eventually at some level of aggregation.

IP flow (IPf) is a stream of packets, sharing the tuple $<$ source IP, destination IP $>$. Defined this way, the flow includes only one direction of traffic and for example a TCP connection will result in two IP flows.

Malicious aggregate (MALagg) is an aggregate of traffic, defined as a set of one or more IPf that, according to a detection mechanism, contains malicious traffic. It should be noted that due to the coarse granularity of its definition, such an IPf can also contain legitimate traffic.

Monitored aggregate (MONagg) is a set if one or more IPf as observed by a monitoring system. Depending on the monitoring system's configuration, it reports MONaggs with a particular granularity, eventually aggregating source and/or destination addresses. In other words, MONagg are defined by the source and destination network addresses (both using CIDR), where the source and destination netmasks are fixed.

Rule denotes an aggregate of source IPs, one destination IP, and an action for the matching traffic. In our case, the action is always deny, i.e., packets matching a rule are dropped. We call the tuple $<$ network source prefix, destination IP $>$ of a rule a filtered aggregate (FILagg). Note that we use source prefix aggregation as explained in Section V-A.

Access Control List (*ACL*) is a set of rules against which traffic is matched by the filtering mechanism. Network equipment often implement ACLs in hardware [8], for performance reasons. On the other hand, hardware implementation becomes with size constraints, and we denote the maximal number of rules in an ACL with N.

B. Threat Landscape

Volumetric DDoS (i.e., bandwidth-depleting DDoS) attacks aim at disrupting a service by consuming the incoming bandwidth and causing congestion at the target, or upstream from the final target.

From the victim (i.e., the final target or a congested network) point of view, the attack sources can appear either as spoofed or not. This means that the malicious traffic's source addresses are faked or real. In fact an attacker can, in some cases, falsify the source IPs of the traffic. In this work, we only consider non spoofed attacks. ghis is a reasonable statement for at least two reasons.

Considering amplification DDoS attacks, which represent a large portion of volumetric DDoS attack [9], massive part of traffic (i.e., from amplifiers towards target) is unspoofed. In fact, the source IP addresses match the sources of traffic, i.e., the amplifier's IPs. Consequently, the number of sources seen in the attack is limited by the number of amplifiers the attacker can find and abuse.

In addition, current direct attacks using Internet of Things (IoT) botnets pave the way to the use of protocols that required non spoofed IP addresses. That is the case of the attack against the Krebsonsecurity website [10] for which attackers made use of the GRE protocol. Remarkably, some direct massive attacks do not make use spoofed traffic, such as the one that hit OVH [11]. The accumulated volume of malicious traffic at each of its network entry points reached around 1Tbps. More than 145k simultaneous non spoofed sources (particularly IoT devices) have been identified as participant of this attack.

C. Assumptions

A prerequisite for blacklisting is the identification of the items to block. In networking, an item refers to network traffic, which can be identified with header fields such as IP addresses, layer 4 protocol and ports. While the detection of the attack is not in the scope of this paper, we expect to obtain alerts containing an exhaustive list of IPfs, i.e., $<$ source IP, target IP $>$ tuples associated with the attack. We consider that this IPf's granularity is a trade-off between the network requirements and mitigation capabilities. In fact, it is coarse enough to be reasonable assumption for the majority of network operator. Besides, it can be regarded as acceptable, in regard to the mitigation, as Pack et al. [6] stated that ACLs can be used as a coarse pre-filter in combination with a finer grained mitigation, such as a middle-box. The middle-box could then trigger an alert using DOTS [12] or IDMEF [13] formats, so that it will include the identification of MALaggs.

III. State of The Art

Filtering traffic is an essential function in a network to mitigate attacks with distributed sources. While some researchers build workarounds to network equipment limitations, the network industry improves the implementation of Access Control Lists in off-the-shelf equipments. This section first provides a review of traffic filtering methods and then we detail the use of ACLs from academic and industrial points of view.

A. Traffic Filtering

Middle-boxes, as proposed for example by Tan et al. [14], aim at providing traffic filtering functions that routers do not implement. Generally, these functions allow a finer-grained filtering and/or are dedicated to mitigate a particular threat. However, the use of a middle-boxe against volumetric DDoS attacks often shifts the bottleneck from the target to the middle-box. Indeed, the attack traffic converging towards the middle-box is likely to cause saturation on the box's upstream link. Qazi et al. [3] studied the deployment of such middle-boxes to address this particular drawback and to dynamically manage the mitigation resources. However, multiplying middle-boxes within the network turns out to be costly.

The use of existing, already in place network equipments to achieve a distributed first line of defense has also been proposed by the industry. Blackholing, such as described by Cisco [15], provides a simple and resource-efficient method [16] to drop a collection of packets based on their destination or source prefix. A destination-based blackhole would, however, disrupt the service by entirely dropping the traffic routed towards it. ACLs can be used for more precise filters, compared to the coarse granularity of blackholing. On routers, ACLs may match on IP header fields, for example source and destination IPs, or the transport layer protocol. Formerly, the major drawback of ACLs was the performance of large ACLs tables. Vendors have fixed this performance issue by implementing filtering in hardware instead of in the router software [17]. The major drawback of the hardware implementation in filtering lists is the limitation of the size of the ACLs [18].

B. Blacklists Implementation and Usage

The use of list of filtering rules, i.e., either whitelists, blacklists or a mix of both, within a constrained environment has been widely studied in the literature. In fact, filtering lists have to be optimized to fit equipment constraints. Industry attempted to solve the CPU consumption issue of software-based filters by implementing them in hardware [17]. However, filtering lists are stored in a fast but expensive memory (TCAM) which is size-limited [6]. Maccari et al. [19] propose the use of a memory efficient structure (Bloom Filter [20]) to reduce the size occupied by a whitelist. [6], [8], [21], [22] considered the memory limitation and aimed at reducing the size of lists using source prefix aggregation.

Several aggregation schemes have been proposed in the literature. Network Aware Clusters [23] identify topologically-closed sources using the BGP routing table. The Hierarchical Heavy Hitters [24] algorithm produces aggregates with approximately equal rates (throughput, bandwidth, etc.) of legitimate traffic. Pack et al. [6] study the capability of filtering lists (whitelists, blacklists and a combination of both) to filter malicious traffic while preserving legitimate traffic. Aggregates are computed using a comparison between a baseline period of traffic and the last period of traffic. Goldstein et al. [21] also propose a history-based algorithm to generate filtering rules using Bayesian decision theory. Soldo et al. [8] develop a framework to build optimal ACLs, where the definition of an optimum depends on the filtering goal, e.g., blocking all sources, some sources, preserving bandwidth, etc. The aggregate computation and selection is driven by weights (i.e., scores) assigned to each source. Although, they evoked a different method to assign scores to sources, the importance of these weights has not been assessed. In this paper, we evaluate the impact of scores based on either flow count or volume, depending on the amount of information about traffic we can retrieve.

IV. Blacklist Generation Problem Characterization

Literature proposes to generate blacklists using the information about the threat and network traffic. However, the attack details depends on the equipment that detect it. Yet, detection mechanisms do not provide equal level of details. Similarly, the visibility that the operator has on his network (e.g., amount of information, level of detail) is highly dependent on the monitoring policy, equipment, etc. We therefore, define three scenarios describing different network visibility levels and illustrate them in Table I.

TABLE I. Information availability-driven scenario

Scenario	Description	MALagg identification	MALagg telemetries	MONagg telemetries
1	Minimum requirement	Yes	No	No
2	Enhanced detection	Yes	Yes	No
3	Full network visibility	Yes	Yes	Yes

The *minimum requirement* scenario describes the minimum information required to generate a rule-based (cf. Section II-A) blacklist, i.e., a list of malicious aggregates (MALaggs), cf. Section II-C.

The *enhanced detection* scenario describes the context where an operator has access to a more detailed information about malicious traffic than solely the identification. He may then be able to retrieve metrics for MALaggs, for example from the detection mechanism. These metrics are collected at the same granularity as the detection, i.e., the tuple $<$ source IP, destination IP $>$.

Our *full network visibility* scenario, evoked in Table I, depicts the use of monitoring information to reduce the amount of collateral damages. Monitoring information is provided for monitored aggregates and, as such it does not differentiate legitimate from malicious aggregates. Off-the-shelf mechanisms, such as NetFlow [25], sFlow [26] or IPFIX [27], are able to provide metrics such as volumetries for traffic aggregates (i.e., MONagg). However, because the granularity of MALaggs is defined by the detection system, and since the MONagg granularity depends on the monitoring system configuration, both aggregate granularities are not always the same. We will study the impact of information availability on the blacklist efficiency in view of these scenarios.

V. Proposition

We propose a filtering scheme that deals with the problems raised in Section II-B and the context exposed in Section IV. As the number of rules in a blacklist is limited, the capability of the ACL to filter malicious traffic depends on the ability of the generation process to aggregate malicious flows, so that the amount of filtered attack traffic is maximized. A workaround would be to increase the amount of traffic to be filtering among the whole traffic, e.g., by shortening the length of the rules source prefix. However, this also probably induces more collateral damage. Consequently, the ACL generation should also tend to minimize the false positives, i.e., legitimate traffic that is being included by the ACL. For example, a DNS server used by the target may also be abused in an amplification attack.

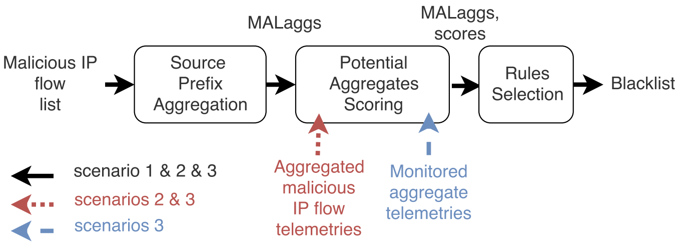
MALaggs
MALaggs, scores
Malicious IP flow list
Source Prefix Aggregation
Potential Aggregates Scoring
Rules Selection
Blacklist
Aggregated malicious IP flow telemetries
Monitored aggregate telemetries
scenario 1 & 2 & 3
scenarios 2 & 3
scenarios 3

Fig. 1. Workflow of ACL generation

The ACL generation process is depicted in Figure 1. First, we compute all possible malicious aggregates (MALaggs) for IP flows IPf included in the alert (cf. Section V-A). Traffic to each destination IP address is treated separately, so that regardless of the filtering granularity, i.e., either based on source IP or both source and destination IPs, only the source IP of malicious flows is aggregated. Second, a score is computed for each of the MALagg (cf. Section V-B) by using aggregated malicious IPf telemetries if they are available (scenario 2 and 3) and monitored aggregates (MONagg) telemetries in scenario 3. Third, the top N MALagg are selected as rules to form the blacklist (cf. Section V-C). Then, the scores are regularly recomputed to maintain up-to-date blacklists, for example every time an alert is received from the detection system.

A. Source Prefix Aggregation

As widely approved by the literature ([7], [21], [24], [28], [29]), we reduce the number of ACL rules using aggregation of source IP addresses for a given destination IP. We thus maintain a separate list of sources for each target.

We define the *aggregation limit* (AL) as the minimal source prefix length of potential aggregates, so that aggregates have a source prefix length between the AL and 32. Figure 2 shows an example of computed source aggregates for a given destination IP. Considering an AL of 26, an alert that contains the following source IPs [1.66.180.12, 1.66.180.13, 1.66.180.50, 1.66.180.60, 1.66.180.201] for a single target results in the following list of possible source aggregates [1.66.180.12/32, 1.66.180.13/32, 1.66.180.50/32, 1.66.180.60/32, 1.66.180.201/32, 1.66.180.12/31, 1.66.180.48/28, 1.66.180.0/26], shown in green in Figure 2. The aggregate 1.66.180.0/24 is not included in the MALaggs as the netmask length exceeds the AL.

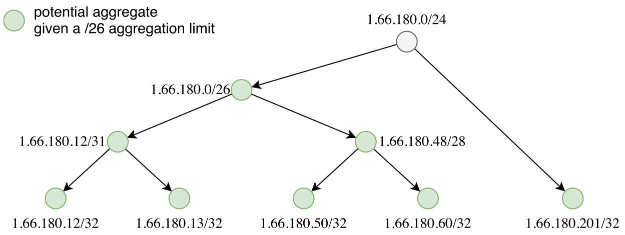

Fig. 2. Example of source aggregation tree for a given destination

B. Malicious Aggregates Scoring

We define three main strategies, for scoring MALaggs that aim at dealing with scenarios that only include information about malicious traffic (cf. scenarios 1 and 2, Table I). A fourth strategy, concerns the last scenario that includes monitoring telemetries. For all strategies, aggregates with high scores are more likely to be added to the ACL. We do not claim that these simple strategies are better than the state of the art. They aims at reflecting how network information can be used and how level of information impacts the filtering.

Scenario 1.a aggregate scoring is relevant to scenario 1 where network operators can only retrieve a list of MALaggs. The generation scheme scores possible aggregates by only taking into account the length of the aggregate's source IP prefix p, as shown in (1). Aggregates with a shorter source IP prefix length get a higher score and are more likely to be inserted in the ACL, such that scores are narrowed between 0 and 32.

$$score_{1.a}(p) = 32 - length(p) \tag{1}$$

Scenario 1.b aggregate scoring also focuses on scenario 1 where operators only get a list of malicious contributors. The score is equal to the ratio between the number of malicious sources ($\mathcal{MS}$) within an aggregate and the complement to 32 of the aggregate's source netmask length, to which has been added 1 so that /32 prefixes does not result in a division by 0. In that case potential aggregates which include larger number of malicious sources and/or whose source prefix is small get a higher score to be put first in the blacklist, so that we try to minimize collateral damages. As a result, score rated between 0 and $2^{32} - 1$ is expressed in (2). In fact, as scored prefixes always contain malicious traffic null score is never reached.

$$score_{1.b}(p) = \frac{|\mathcal{MS} \cap p|}{(32 - length(p)) + 1} \tag{2}$$

Scenario 2 aggregate scoring also takes the malicious IPf telemetries as input. Since we aim at mitigating volumetric attacks and their congestion effect on the network, we consider the volumetry as the ground metric to assess the impact of aggregates. The aggregate score - expressed in bytes in (3) - refers to the volume of malicious traffic towards the target, which source IPs ($\mathcal{MS}$) are included in the aggregate source prefix p. The $malVol(ip)$ function depicts the byte sum of malicious traffic from ip towards the target reported during the last period. As such, the score ranges from 0 to the total volume of attack traffic.

$$score_2(p) = \sum_{ip \in \mathcal{MS} \cap p} malVol(ip) \tag{3}$$

Scenario 3 aggregate scoring depicts the use of MONagg telemetries. MALagg's scores - also expressed in bytes and ranged from 0 to sum of volumes of all malicious IPfs - are obtained by multiplying the score obtained in scenario 2 and the ratio between the volume of malicious traffic and an estimation of the overall traffic within the aggregate p ($overallVol(p)$), as expressed in (4).

$$score_3(p) = score_2(p) \times \frac{score_2(p)}{overallVol(p)} \tag{4}$$

This is an estimation because the length of source prefix of potential rules may not always be equal to the prefix length of source prefix of monitored aggregates . In fact, the length of source prefix (p) of the potential aggregates to filter varies between the aggregation limit and 32, while the source prefix (p_m) length of MONagg is fixed by the monitoring system configuration (cf. Section IV). The estimation depends on the value of the source prefixes' length as can be seen in (5).

$$overallVol = \begin{cases} \sum_{p_m \subset p} monVol(p_m) & \text{for } length(p) < length(p_m) \\ monVol(p) & \text{for } length(p) = length(p_m) \\ (monVol(p_m) - malVol(p)) & \\ \times \frac{nbHosts(p)}{nbHosts(p_m)} + malVol(p) & \text{otherwise} \end{cases} \tag{5}$$

If the prefix of a MONagg is larger than the prefix to filter, the estimation of is equal to the sum of the volume of all monitoring aggregates ($monVol$) included in the source prefix to filter p, The estimation is equal to the volume of the monitoring aggregate when the length of the aggregate to filter is equal to configured prefix length of monitoring aggregates. Otherwise, we estimate the volume of legitimate traffic within the source prefix p towards a given destination. We first assume that remaining traffic volume (i.e. $monVol(p_m) - malVol(p)$) is evenly distributed on the highest number of hosts in a subnet of size $length(p_m)$ expressed in (6). Then, we add the volume of these sources included in the prefix p to the volume of malicious traffic for this aggregate.

$$nbHosts(p_m) = 2^{32-length(p_m)} \tag{6}$$

C. Rules Selection

Finally, we select the top N rules among all scored potential aggregates to form the blacklist. The process is depicted in Figure 3. The potential aggregates are sorted according to their scores in descending order. In the example, scores between parentheses have been set arbitrarily. However, it is possible that the aggregate $1.66.180.12/32$ has a higher score than one of its parent aggregate, e.g., $1.66.180.12/30$. A legitimate client (e.g., $1.66.180.14$) with a large volumetry may reduce the $1.66.180.12/30$ aggregate score. Then, the top N ($N = 3$ in Figure 3) are used to generate the ACL. Considering the aggregation mechanism, an overlap is possible only if an aggregate is included another. Consequently, to avoid wasting rules, an ACL has to be exclusive. Then, when we try to insert in the top N an aggregate that includes or is included in an already inserted aggregate, we keep the aggregate with the smallest prefix length and remove the other. In the example, $1.66.180.50/32$ and $1.66.180.48/28$ are both in the top 3 aggregates. However, as the second aggregate includes the first one, only $1.66.180.48/28$ is kept in the final blacklist.

VI. Experiments

Blacklists are widely used to mitigate DDoS attacks. However, while literature has proposed algorithms to generate such filters with a realistic number of rules, they do not evaluate the efficiency of their approach in regard of amount of information on the network available. These proposed algorithms, however, are not applicable in all networks due to the requirements in terms of information availability. In this paper, we formulated three scenarios describing different network visibility levels and proposed basic blacklist generation strategies for each scenario. We conduct simulation in which we generate blacklists using scenario related strategies and apply resulting filters on traffic captures. We then compare results for each scenario-driven strategy. It allows us to study the impact of the levels of available information on the filtering efficiency, i.e., the ability to drop malicious traffic while preserving legitimate flows.

A. Metrics and Variables

We rely on two commonly used metrics when dealing with filtering, the *true positive rate* (TP_r, also known as sensitivity or recall) and the *false positive rate* (FP_r) in order to assess the scoring strategies. The TP_r evaluates the proportion of malicious traffic that is being filtered by the mechanism, how the filter is able to correctly drop malicious traffic. The generation strategies have been designed to maximize this percentage. Conversely, the FP_r measures the proportion of collateral damages. This allows validating the use of monitoring information to reduce the collateral damages. Both metrics are then well fitted to assess the twofold definition of the efficiency.

The TP_r is obtained as the ratio between the number of filtered malicious IPf and the total amount of malicious flows. Correspondingly, the false positive rate (FP_r) is the ratio between the number of filtered legitimate flows and the sum of legitimate flows.

The blacklist construction is tuned using two parameters, the maximal number of rules in a filter (N) and the aggregation

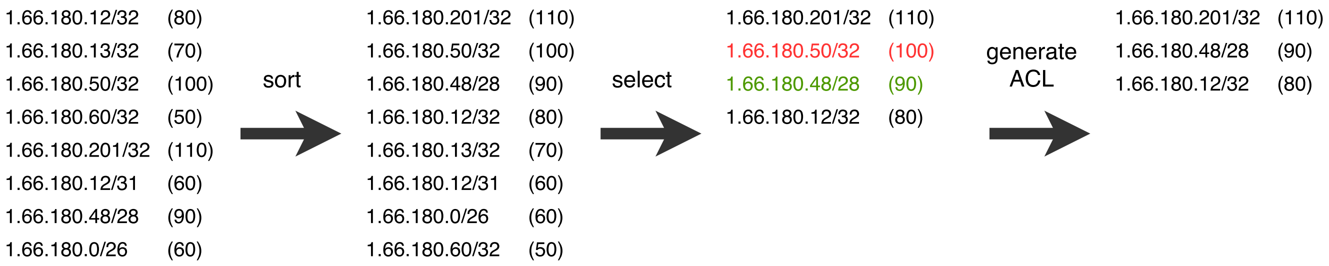

Fig. 3. Selection of top 3 rules among potential source aggregates for a given destination

limit (AL). Soldo et al. [8] used from few hundreds to few thousands rules. We then execute experiments with different values of N between 10 and 500 maximum filtered aggregates in the filter. Aggregation limit is fixed to either /24 or /8 to study its impact with a short and a long filtered aggregates prefix length. An AL of /0 and /32 have not been considered here, as a filtered aggregate /0 will result in dropping the whole traffic. Conversely, filtering with the whole IP (/32 prefix) instead of an aggregate, with at most 500 rules cause at most 0.25% of malicious IPf to be filtered which we can consider as pointless. In fact, this depends on how aggressive are these IPfs. However, for reasons of clarity, we decided not to include /32 prefixes.

We consider two configurations for the MONagg's granularity reported by the monitoring system. The first one define records for destination IPs, i.e. traffic metrics are reported for $<$ /0 source prefix, /32 destination prefix $>$ aggregates. This kind of monitoring configuration may be used for networks where each customer is identified by the destination IP such as data centers. The second finer granularity is defined by the tuple $<$ /24 source prefix, /32 destination prefix $>$. That can be used, for example, by ISPs, so that records match the largest common inter-AS BGP prefixes advertisements [30].

B. Results

The behavior of ACL-based filtering, as described in Section V-B, is studied for each scenario defined in Section IV. We use real legitimate traffic from the MAWI data set [31] as legitimate traffic superimposed with generated attack traffic. Traffic has been captured on February 2017 during 15 minutes on a transit link and has been cleaned from attack traffic [1]. The inbound part of this capture has an average packet rate of 51,000 packet per second (295 Mbps). In parallel, we generate 10 different attack traffics.

In order to consistently run experiments with the MAWI capture and one of the 10 attack traffic, we follow the procedure below. We select 1000 legitimate sources from the MAWI capture that will also send attack traffic. The remaining malicious sources are randomly chosen such that they are not seen in the legitimate capture. In total, 200,000 malicious sources are selected. We generated a constant bit rate attack traffic with a bandwidth of 1.3Gb/s. While the overall number of attack sources is realistic [10], [11], the overall volume fall short of the most massive current attacks due to computational constraints. Each of the MALagg has also a constant throughput throughout the attack, which is randomly chosen between 0.6 and 1.4 the average per flow bit rate. More realistic source dynamics will be considered in future works. Scores are regularly re-computed (e.g. every 60 seconds in Table II) to update the variation of legitimate traffic. Figure 4 shows the average true and false positive rates (TP_r and FP_r) for each scoring function with multiple ALs (depicted in columns) and varying values of N (x-axis).

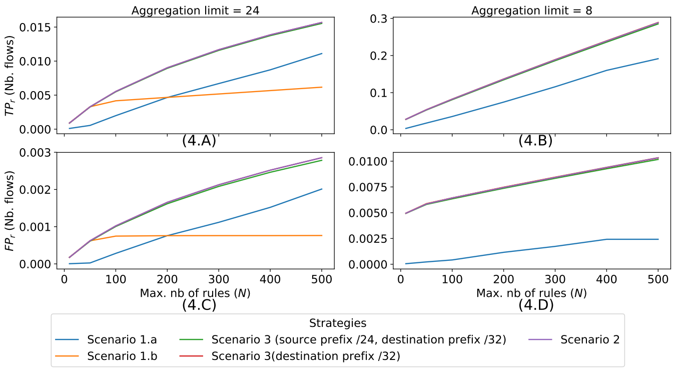

Fig. 4. Comparison of scoring functions

1) Scenario 1 - Minimum Requirements: For the *Scenario 1.a* score function, the TP_r shows odd trends. For example, in Figure C, the TP_r has a small increase for N varying from 10 to 50. This growth increases for a number of rules greater than 50. In fact the strategy does not succeed in selecting the top N rules. This is due to the fact that a lot of filtered aggregates (more than 190,000) have the same score. However, the score function has to select the top N, where N is less than 500. There is therefore no rational method to select the top N rules. This results in a pseudo random selection of filtered aggregates .

The *Scenario 1.b* scoring function (in orange) grows linearly from 10 to 50 rules in the filter for each sub-figure. In fact, the number of malicious IPf added per rule linearly decreases considering a maximum rule count greater than 50. However, the TP_r of *Scenario 1.b* score function shows a

[1] MAWI capture is available at http://www.fukuda-lab.org/mawilab/v1.1/2017/02/03/20170203.html

much larger increase for $N = 10$ to 50. This is also due to the distribution of traffic, where the top N rules contain very dense malicious aggregates with significantly high volumetry. The *Scenario 1.b* scoring function is then correctly choosing the top 10 to 50 rules. This large increase for small N values does not appear in sub-figure 4.B, as malicious sources are more evenly distributed among the source aggregates for small prefixes. In other words, in Figure B, traffic aggregated in /24 source prefixes emphasizes some aggregates with high impact. However, these /24 aggregates with high score are diluted in /8 aggregates.

C. Scenario 2 - Enhanced Detection

The TP_r and FP_r of the *Scenario 2* scoring function (in violet) coincides with the *Scenario 1.b* from $N = 10$ to 50 in Figures A and C. Using malicious IPf telemetries provide no added value given a small number of rules in a filter. Conversely, from $N = 50$ to 500 in Figure A, the TP_r of the *Scenario 2* grows faster than for the baseline scenario. For example, given $N = 500$ (cf. Table II), the number of dropped malicious flows in *Scenario 2* is just over twice for the *Scenario 1.b*, and the same proportion applies considering the filtered volumetry. The drawback is that the FP_r also grows faster (cf. Figure C), resulting in an increase of the collateral damages. Table II shows that, for $N = 500$, the legitimate dropped traffic in *Scenario 2* is around 4 and 5 times the *Scenario 1.b* statistics expressed respectively in terms of number of flows and volumetry. Both behavior are mostly due to the fact that the scores of *Scenario 1.b* are devalued when the source prefix of the MALagg grows, that also induces an increase of the probability to include legitimate traffic. The *Scenario 2* does not try to reduce the collateral damages. However, the chosen scenario scoring functions is not so efficient with an AL of /24, as only 1.50% of malicious traffic is filtered.

When we shorten the AL, e.g., from Fig A to B, the *Scenario 2* and *1.b* display similar trends. In fact, a large part of malicious IPf is quite dispersed, so that long source prefixes only aggregate a small part of malicious traffic. As a consequence, in Figure A, the *Scenario 1.b* scoring function favor malicious flows over malicious aggregates. In contrast, shorter source prefix (i.e. up to /8) aggregates more malicious traffic, so that they get by the *Scenario 1.b* a higher score than longer prefix aggregates. As a consequence, filtered aggregates selected by the *Scenario 1.b* scoring function matches the ones selected by the *Scenario 2* and the FP_r curves coincide, cf. Figure B. Both *Scenario 1.b* and *2* scoring functions allow filtering around 30% of malicious traffic (in terms of number of flows and volumetry), using solely 500 rules with an average of 1% of filtered legitimate traffic (Table II).

1) Scenario 3: The curves of *Scenario 3* scoring function configured with a /32 destination prefix granularity (depicted in red) coincide for all sub-figures with the results of the *Scenario 2*. Considering an AL of /24 (Figure A), this is due to the fact that very few filtered aggregates contain both legitimate and malicious traffic, so that introducing monitoring information is not been able to provide much value-addition. For an AL of /8, cf. Figure B, the reason is that legitimate and malicious traffics are highly distributed among filtered aggregates , so that scores get similar results for the rules. This also explains the fact that the FP_r trends of the *Scenario 3* configured with $<$ /24 source prefix, /32 destination prefix $>$ granularity (shown in green) results almost similar to the *Scenario 2* false positive rate. While this means that it does not reduce the efficiency of the blacklist in terms of malicious traffic filtering, this does not help in preserving legitimate traffic.

The *Scenario 3* that uses$<$ /24 source prefix, /32 destination prefix $>$ MONagg granularity, depicts an improvement of the FP_r compared to the *Scenario 2* in Figure C. Although this seems small when expressed in terms of number of flows (i.e., 0.01% of legitimate flows preserved, cf. Table II), results are a little more significant when the FP_r is expressed in terms of volume (0.2%).

D. Discussion

Experiments with attack traffic without variations show the basic efficiency and behavior of scoring functions for the considered scenarios. First, considering the hardware limitations of the number of rules in a router, the *Scenario 1.a* scoring function is irrelevant. In fact, the strategy does not allow choosing correctly the top N rules, as more than N rules obtain the best score. In order to use it effectively, routers would require at least around 190k rules for an AL of /24. This minimum number of rules is dependent on the malicious traffic distribution, i.e., poorly distributed malicious traffic would require fewer rules to be aggregated. The *Scenario 2* reaches the highest efficiency when it comes to only filtering malicious traffic. However, as we increase the AL, the *Scenario 1.b* scores results similar to the *Scenario 2*. In other words, the optimal efficiency in the scenario 1 is at the same level as the efficiency of a scenario with a higher level of detail (scenario 2), assessed in terms of the number of dropped flows.

Our evaluation is based on the assumption that the malicious traffic is not spoofed, as explained in Section II-B. If it were not the case, DDoS mitigation with blacklists could cause more collateral damage, as spoofed attack traffic could overlap more easily with legitimate traffic and increase the number of MALagg that also contain legitimate traffic. We configured an IPf-level overlap of 1,000 over 200,000, which seems in most cases far above reality. For example, considering amplification attacks, this means that the target legitimately connect with 1,000 amplifiers (DNS servers, . . .). This may be the case when the target is a proxy or a NAT gateway. However the overlap is exacerbated using source aggregation.

In our approach, we also supposed that the list of malicious aggregates, e.g., contained in the alert, is exhaustive. This is not true in all cases, as the detection mechanisms are not perfect and/or do not report attack sources exhaustively due to the potentially large number of sources in DDoS attacks.

TABLE II. Average statistics of dropped traffic for each scenario considering a blacklist generation each 60s ($N = 500$)

			Scenario 1.a	Scenario 1.b	Scenario 2	Scenario 3 (source prefix /24, destination prefix /32)	Scenario 3 (destination prefix /32)
AL = /24	malicious traffic	#IPfs	2222 (1.11%)	1234 (0.62%)	3136 (1.57%)	3107 (1.55%)	3136 (1.57%)
		std	50	55	25	28	25
		MBytes	108.5 (1.11%)	60.28 (0.62%)	154.89 (1.59%)	153.68 (1.58%)	154.89 (1.59%)
		std	2.63	2.81	1.38	1.47	1.38
	legitimate traffic	#IPfs	1222 (0.2%)	462 (0.08%)	1730 (0.29%)	1687 (0.28%)	1729 (0.29%)
		std	814	266	1010	2012	1010
		MBytes	8.06 (0.37%)	2.89 (0.13%)	14.68 (0.67%)	10.45 (0.48%)	14.67 (0.67%)
		std	1.73	0.91	3.42	2.69	3.42
AL = /8	malicious traffic	#IPfs	38291 (19.15%)	57822 (28.91%)	57722 (28.86%)	57020 (28.51%)	57713 (28.86%)
		std	164	168	172	226	171
		MBytes	1866.1 (19.14%)	2819.38 (28.92%)	2824.54 (28.97%)	2790.21 (28.62%)	2824.02 (28.96%)
		std	8.34	8.67	8.48	11.18	8.46
	legitimate traffic	#IPfs	1437 (0.24%)	6195 (1.03%)	6191 (1.03%)	6096 (1.02%)	6182 (1.03%)
		std	29	1657	1656	1665	1657
		MBytes	0.22 (0.01%)	32.16 (1.47%)	32.16 (1.47%)	32.02 (1.46%)	31.62 (1.45%)
		std	0.01	5.25	5.23	5.03	5.27

However, to be efficient, a detection mechanism is likely to provide top malicious aggregates, i.e., the aggregates with most impact, e.g., the aggregates with the highest data rate as we deal with volumetric DDoS. The malicious aggregates not included in the alert are thus likely have only a small effect in the network congestion.

We focused in this paper continuous per IPf throughput. The impact of more dynamic attack traffic will be studied later. We expect that the efficiency would be affected by new parameters such as the monitoring records collection period, and the blacklist refresh period. Moreover, history-based algorithm should be studied, such as Exponentially Weighted Moving Average (EWMA) used in [7], [32], to generate blacklists which handle temporal trends of malicious IPf contributions.

VII. Conclusion

We presented an evaluation of simple blacklisting algorithms, from the perspective of an operational constraint, i.e., level of information an operator can retrieve from the network. The assessment scenarios considered the fact that operators do not have equal level visibility on their networks, depending on the functionality and configuration of the monitoring system. The minimal requirement for blacklist generation is the identification of source IPs of attack traffic. From there on, additional traffic information, such as volumetries of attack and legitimate traffic, can be used to improve the efficiency of the blacklists. Experiments highlighted that a generation algorithm does not fit well in all scenarios and it should be carefully chosen in regard of the available network information. Furthermore, in some situations providing more detailed information improved the filtering results only up to a given point, suggesting that the algorithms' behaviors should be evaluated in the context in which they are to be used.

We also considered in this paper the aggregation of monitored traffic (i.e., generation of monitored aggregates) as a possible optimization of the monitoring system, although it will degrade the quality of flow reporting. We acknowledge that this is not the only possible configuration parameter a network operator is able to leverage to optimize network monitoring system, for example flow sampling is another widely used optimization. While the impact of flow sampling on the attack detection [33], [34] has been studied, its effect could also be studied in regard of the scenario 3. As a consequence, efficiency of filtering can be assessed in the light of the cost of monitoring collection, in term of storage, flow records bandwidth consumption.

References

[1] M. Casado, P. Cao, A. Akella, and N. Provos, "Flow-Cookies: Using Bandwidth Amplification to Defend Against DDoS Flooding Attacks," in *Proc. IWQoS*, jun 2006, pp. 286–287.

[2] A. Greenhalgh, M. Handley, and F. Huici, "Using Routing and Tunneling to Combat DoS Attacks." in *SRUTI*, 2005, pp. 1–7.

[3] Z. A. e. a. Qazi, "SIMPLE-fying Middlebox Policy Enforcement Using SDN," in *ACM SIGCOMM*, 2013, pp. 27–38.

[4] A. Mahimkar, J. Dange, V. Shmatikov, H. M. Vin, and Y. Zhang, "dFence: Transparent Network-based Denial of Service Mitigation." in *4th USENIX Symposium on Networked Systems Design & Implementation (NSDI 07)*, 2007, pp. 327–340.

[5] A. Abujoda and P. Papadimitriou, "MIDAS: Middlebox discovery and selection for on-path flow processing." in *COMSNETS*, 2015, pp. 1–8.

[6] G. Pack, J. Yoon, E. Collins, and C. Estan, "On Filtering of DDoS Attacks Based on Source Address Prefixes," in *Securecomm*, 2006, pp. 1–12.

[7] F. Soldo, A. Le, and A. Markopoulou, "Predictive blacklisting as an implicit recommendation system," in *2010 Proceedings IEEE INFOCOM*, March 2010, pp. 1–9.

[8] F. Soldo, K. Argyraki, and A. Markopoulou, "Optimal Source-Based Filtering of Malicious Traffic," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 381–395, apr 2012.

[9] "Verisign Distributed Denial of Service Trends Report - 3rd Quarter 2017," Tech. Rep. 3, 2017.

[10] B. Krebs, "Krebsonsecurity hit with record ddos," 2016.

[11] OVH, "The DDoS that didn't break the camel's VAC," 2016.

[12] N. Teague, "Open threat signaling using rpc api over https and ipfix," Internet-Draft, July 2015.

[13] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)," RFC 4765 (Experimental), IETF, Mar. 2007.

[14] T. H. Tan, C. Y. Ooi, and M. N. Marsono, "rrBox: A Remote Dynamically Reconfigurable Middlebox for Network Protection." in *CANDAR*, 2014, pp. 219–225.

[15] "Remotely Triggered Black Hole Filtering - Destination Based and Source Based," Cisco Systems, Tech. Rep., 2005.

[16] I. Vordos, “Mitigating Distributed Denial of Service Attacks with Multi-Protocol Label Switching-Traffic Engineering (MPLS-TE),” Ph.D. dissertation, Naval Postgraduate School, 2009.

[17] “Understanding ACL on catalyst 6500 series switches,” Cisco, Tech. Rep.

[18] M. Xu, S. Yang, D. Wang, F. Li, and J. Wu, “Source address filtering for large scale networks,” *Computer Communications*, pp. 64–76, 2014.

[19] L. Maccari, R. Fantacci, P. Neira, and R. M. Gasca, “Mesh Network Firewalling with Bloom Filters,” in *ICC*, 2007, pp. 1546–1551.

[20] B. H. Bloom, “Space/Time Trade-offs in Hash Coding with Allowable Errors.” *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[21] M. Goldstein, C. Lampert, M. Reif, A. Stahl, and T. Breuel, “Bayes Optimal DDoS Mitigation by Adaptive History-Based IP Filtering,” in *ICN*, 2008, pp. 174–179.

[22] A. Kalliola, T. Aura, and S. Šćepanović, “Denial-of-Service Mitigation for Internet Services,” in *Proc. NordSec*, 2014, pp. 213–228.

[23] B. Krishnamurthy, J. Wang, B. Krishnamurthy, and J. Wang, “On network-aware clustering of Web clients,” *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 97–110, oct 2000.

[24] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava, “Finding Hierarchical Heavy Hitters in Data Streams,” in *VLDB*, vol. 29, 2003, pp. 464–475.

[25] “NetFlow,” Cisco, Tech. Rep., 2008.

[26] “Traffic Monitoring using sFlow,” sFlow.org, Tech. Rep., 2003.

[27] G. Sadasivan, N. Brownlee, B. Claise, and J. Quittek, “Architecture for IP Flow Information Export,” RFC 5470, IETF, Mar. 2009.

[28] A. Kalliola, K. Lee, H. Lee, and T. Aura, “Flooding DDoS mitigation and traffic management with software defined networking,” in *Proc. CloudNet*, 2015, pp. 248–254.

[29] M. Collins and M. Reiter, “An Empirical Analysis of Target-resident DoS Filters,” in *Proc. SECPRI*, 2004, pp. 103–114.

[30] D. Bayer, “Visibility of Prefix Lengths in IPv4 and IPv6,” 2010. [Online]. Available: https://labs.ripe.net/Members/dbayer/visibility-of-prefix-lengths

[31] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, “MAWILab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking.” in *CoNEXT*, 2010, p. 8.

[32] J. Freudiger, E. De Cristofaro, and A. E. Brito, “Controlled data sharing for collaborative predictive blacklisting,” in *Detection of Intrusions and Malware, and Vulnerability Assessment*, M. Almgren, V. Gulisano, and F. Maggi, Eds. Cham: Springer International Publishing, 2015, pp. 327–349.

[33] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, “Impact of packet sampling on anomaly detection metrics.” in *Internet Measurement Conference*. ACM, 2006, pp. 159–164.

[34] A. Pescape, D. Rossi, D. Tammaro, and S. Valenti, “On the impact of sampling on traffic monitoring and analysis,” in *ITC*, 2010, pp. 1–8.