Deciphering DDoS Attacks through a Global Lens

Jonas Brunner[1], Bruno Rodrigues[1], Katharina O. E. Müller[1], Salil S. Kanhere[2], Burkhard Stiller[1]
[1]Communication Systems Group CSG, Department of Informatics IfI, University of Zurich UZH, Switzerland
[2]Networked Systems and Security Group NetSyS, UNSW Sydney, NSW 2052 Australia
E-mail: jonas.brunner@uzh.ch, [rodrigues|mueller|stiller]@ifi.uzh.ch
salil.kanhere@unsw.edu.au

***Abstract*—With a rising frequency and scale, Distributed Denial-of-Service (DDoS) attacks persist as a critical cybersecurity issue. While shared attack fingerprints aid many intrusion detection systems in identifying threats, their application for DDoS attacks remains limited due to their distinct nature. However, fingerprints observed from multiple locations can provide valuable insights. This paper presents Reassembler, a novel platform for achieving a global DDoS attack analysis using attack fingerprints recorded from various locations. Reassembler consolidates these fingerprints into a unified view allowing to obtain a global overview of DDoS attacks. The evaluation, conducted on four simulated scenarios, demonstrates Reassembler's ability to extract novel properties, such as the count of intermediate nodes and the estimated percentage of spoofed IPs.**

***Index Terms*—Distributed Denial-of-Service, Attack Fingerprints, Cooperative Defense**

I. Introduction

Distributed Denial-of-Service (DDoS) attacks significantly threaten Internet availability and digital security in today's highly connected world. Over time, DDoS attacks have evolved, not only in their accessibility (*e.g.*, using stress-testers or Booters [22]) but also in terms of bandwidth consumption. Reports of attacks exceeding 1 Tbps in bandwidth, as noted by Cloudflare, one of the leading DDoS mitigation companies, serve to underscore this reality [21]. Furthermore, attackers are now leveraging DDoS attacks for ransomware activities, further compounding the threat these attacks pose to businesses and individuals alike [11]. However, despite numerous efforts in both commercial and research realms, DDoS attacks remain unsolved.

Effective counter-strategies to these distributed attacks necessitate an equally distributed defense mechanism, preferably one that mitigates attacking traffic at different points, potentially closer to their origins [17], [18], [9]. Information sharing is critical to successfully implementing this approach, which can enhance the distributed DDoS attack defense. This has motivated the use of fingerprints or signatures for sharing attack data and patterns in various Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) [14]. The DDoSDB project [7], for example, was proposed as a central repository where organizations can share their DDoS fingerprints, thereby expanding the collective knowledge on the subject. However, the distributed nature of DDoS attacks presents unique challenges in analyzing and aggregating such fingerprints [22].

Historically, DDoS defense strategies have primarily been implemented at the attack target [10], [16], [12]. This is largely due to the challenges in universally applying a specific DDoS attack fingerprint to detect similar attacks. However, the distributed nature of DDoS attacks provides a unique advantage: they are also observable by multiple network nodes. While sharing mechanisms for DDoS attack fingerprints currently exist [7], they have seldom been employed to provide a global perspective of these attacks as they only report the target's viewpoint. This paper posits that attack fingerprints recorded at various nodes offer unique insights into DDoS attacks and when aggregated, can provide a novel foundation for a global post-mortem analysis.

To the best of the author's knowledge, no existing work has provided a comprehensive analysis of a DDoS attack based on fingerprints recorded at different locations. Reassembler addresses the challenge of filtering and aggregating DDoS attack fingerprints to form a global analysis, leveraging these observations to derive novel metrics enriching DDoS attack analysis and fingerprinting methods in general. Contributions are summarized as follows:

- **Extended Fingerprint Format** enhancing the ability to capture and share intricate details of DDoS attacks, fostering an improved understanding and enabling more comprehensive analyses of attacks.
- **Topology Generator** an additional tool enabling the creation of realistic, diverse network topologies, essential for simulating and analyzing a wide range of DDoS attack scenarios within Reassembler.
- **Reassembler Tool** that collects and consolidates attack fingerprints from various locations providing a unique, holistic view of DDoS attacks. Reassembler's code is available in [2], [3].
- **Data-driven** approach as Reassembler provides a robust foundation for multi-perspective DDoS post-mortem analysis.

This paper is organized as follows. Section II presents related work. Section III describes Reassembler's design. Section IV presents evaluation results and discussion. Summary and future work are presented in V.

978-3-903176-59-1

II. Related Work on DDoS Fingerprinting

The literature on DDoS attack fingerprinting does not consistently define what a fingerprint means. Lee and Shieh [12] consider the route a packet has traversed between source and destination to be its fingerprint. They propose a filtering scheme that recognizes and drops packets with a spoofed IP address based on their path fingerprint. Osanaiye [16] denotes a fingerprint as the operating system of an attacking device. A combination of active and passive fingerprints detects whether an IP packet is spoofed. The passive fingerprint is generated by analyzing incoming packets; the active fingerprint is by probing the source IP. An IP packet is considered valid if both fingerprints extracted from operating systems match.

Wang *et al.* [20] employ an IP packet hop count as a fingerprint against which future requests are compared to filter out packets with spoofed IP addresses. They infer the hop count from each packet's IP header's Time-to-Live (TTL) field. In the learning state, the *Hop-Count-Filterig* (HCF) technique generates and collects fingerprints. Once packets that do not match the stored fingerprint for that IP are detected, HCF changes to the filtering state and discards those packets.

The fingerprint definition of [10] and [5] is closest to the proposed definition, as the fingerprint represents a standalone view of a distinct DDoS attack. The problem with other presented fingerprinting methods is that they are only a means to detect an attack, but they do not uniquely characterize an attack. As such, they cannot comprehensively analyze a specific DDoS attack. Furthermore, not all methods can be deployed directly but require a change in the underlying Internet protocol (*e.g.*, path fingerprint [12]).

All of the above-mentioned fingerprinting methods are destination-based. They help to detect and mitigate an attack on the target's side. This is certainly important from a victim's perspective. However, more global views of DDoS attacks are also required to mitigate such threats efficiently. In this regard, Akella *et al.* [1] propose a detection method that helps ISP networks detect attacks on themselves and external attacks that use a certain ISP network. Their detection method relies on stream-sampled profiles of normal traffic and applies anomaly detection. Aggregated fingerprint data is then shared among the routers in the ISP networks using a simple consensus mechanism based on a predefined confidence threshold.

Another approach to get a global view of DDoS attacks is analyzing darknet traffic (*i.e.*, routable but unused Internet addresses). This idea was initially proposed in [15] and was based on the assumption that DDoS attackers randomly generate a spoofed source IP for each packet. When observing a large enough IP range (*e.g.*, 1/256 of the IPv4 space), Moore *et al.* [15] were able to sample an attack overview of the whole Internet space using backscatter analysis. Following these footsteps, Fachkha *et al.* [8] extend this idea without relying on backscattered analysis. They conduct a flow-based traffic analysis on DNS queries to the darknet space. Their results show that the global increase of DNS queries of type *ANY* is caused by DNS amplification attacks.

There is no unified definition of an attack fingerprint, and various detection methods employ different fingerprint formats [12], [16], [20], [10], [13]. Also, not all are ready to be deployed, either requiring changes to underlying protocols [12] or collecting data over months up to years [15], [8]. Reassembler builds upon the definition proposed in [10] and [5] as they represent a standalone view of a distinct DDoS attack. Reassembler uses fingerprints not only as an attack description but also as a simple traffic observation and introduces novel properties (*cf.* Extended Fingerprint Format III-1) that allow a global analysis. For example, reporting on the detection threshold, number of packets, and Time-to-Live (TTL) per source. These properties, although not precisely reported at an intermediary node, can present valuable insights when aggregated.

III. Reassembler's Design

Reassembler is a tool for comprehensive post-mortem DDoS attack analysis utilizing attack fingerprints from various locations. Simultaneously, a unique module for generating custom attack scenarios has been developed, fostering the production of realistic attack fingerprints to test and enhance Reassembler's performance. Altogether, Reassembler's contribution lies in its ability to provide a global, holistic view of DDoS attacks by using and aggregating attack fingerprints from various locations.

1) ***Extended Fingerprint Format****:* Different definitions of DDoS fingerprints exist (*cf.* Section II). This paper extends a dedicated program *DDoS Dissector* [6] used to generate fingerprints from network traces (*e.g.*, PCAP files). The Dissector employs a straightforward detection mechanism, identifying IP addresses that received more than 50% of the analyzed traffic. If the 50% threshold is unmet, the Dissector looks for a target subnet.

While this is optimal for analysis at a single point (*i.e.*, at the victim), it poses problems when fingerprints from multiple recorded locations are combined. For global analysis, it is also desirable to have attack fingerprints with less certainty as they can be interesting in a global context. For example, with a high threshold of 50%, only nodes that are very close to the target (and the target itself) report a fingerprint, but with high certainty. Targets that receive less than 50% are discarded, and no fingerprint is generated. With a lower threshold (*e.g.*, <10%), nodes farther away might also produce a fingerprint with lower certainty. While a single fingerprint with low certainty cannot identify or describe an attack, it helps to build a global picture.

To address the difficulty in attributing fingerprints to their recording locations due to the imprecision of IP address and TTL value assumptions, we introduce a `location` property. This property accurately pinpoints where a fingerprint was captured. Figure 1 illustrates an example where Router *R* receives traffic using the same protocol and port from three different sources.

TABLE I: Proposed Additional Properties for Attack Fingerprints (Extending [10])

Level	Field name	Description	Data type
Fingerprint	`location`	The IP of the location where the fingerprint was recorded	String
Attack Vector	`nr_packets_by_source`	Number of packets in the attack vector grouped by source IP	Map<String, Integer>
Attack Vector	`ttl_by_source`	Observed TTLs in the attack vector grouped by source IP	Map<String, Integer[]>
Attack Vector	`detection_threshold`	Percentage of attack traffic compared to all observed traffic	Float

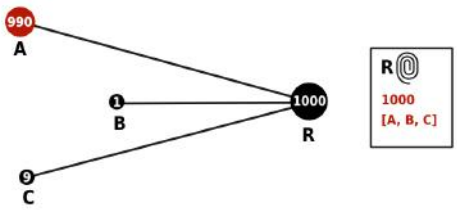

Fig. 1: Loss of Source Context During a Fingerprint Aggregation

TABLE II: Output of the Attack Identification Stage

Parameter	Description	Finality
Attack Target	One single attack target of a globally observed attack	Final
Intermediate Nodes	List of nodes that (potentially) observed the attack	Requires Refinement
Attack Sources	List of attack sources (not filtered for spoofed addresses)	Requires Refinement

Client *A* maliciously sends a lot of packets (*e.g.*, SYN flood), while client *B* and *C* solely try to access the website. Due to the aggregation per port and protocol, router *R* calculates a fingerprint with three attack sources and a total number of 1'000 packets. This means non-malicious source IPs are included in the fingerprint if they use the same protocol and port within the same timeframe. Such behavior can be detected using the reported TTL values to a certain extent. However, if all arriving packets at router *R* have the same TTL, the fingerprint makes no distinction between malicious or non-malicious source IPs. As mentioned before, such properties in the fingerprint are not required when generating a fingerprint at a single location with high certainty. However, the additional information provides an interesting basis for further analysis of a global picture. The additional proposed properties are summarized in Table I.

A. The Reassembler Module

The Reassembler module generates a global overview of a DDoS attack given a set of attack fingerprints recorded at different locations. This is crucial to identify patterns and efficiently develop countermeasures that cannot be derived from observing the DDoS attack at a single point. For example, the number of intermediate nodes and their level of involvement in an attack or the number of intermediate nodes helps to understand an attack's scale and assess attackers' resources and strategies. The Reassembler can be structured into four major steps.

1) ***Pre-process Fingerprints****:* Pre-processing attack fingerprints includes two main actions:

- **Read/Load Attack Fingerprints** from a shared location (*e.g.*, an online repository) Since this paper uses a simulated scenario, the shared location is emulated by a local folder.
- **Convert Fingerprints** to a format allowing further processing. To make the lookup and filtering of data efficient, fingerprints are converted to a flat (*i.e.*, non-nested) data format.

While one could argue that loading attack fingerprints falls outside the pre-processing stage, it is included as the purpose of pre-processing is to provide the data for the next step such that it can be analyzed efficiently.

2) ***Attack Identification****:* An attack fingerprint describes a specific attack generated by a previously identified attack [10]. This assumption is weakened so that a fingerprint does not necessarily describe an attack but serves as a mere traffic observation. On the one hand, this removes the complex logic and computational overhead of identifying an attack from the network nodes. On the other hand, it requires identifying attacks later by looking at a large set of possible attack fingerprints.

Considering a case where many nodes in the network contribute their recorded attack fingerprints in a shared repository, there are different ways to infer an attack target. Three possible scenarios arise in DDoS attack reporting. First, a node actively signals to be an attack target and provides a detailed fingerprint. In the second, there is no active declaration of being a target, but fingerprints are submitted to a shared repository as a background operation of the DDoS Dissector [6]. In the last, the attack target neither announces the attack nor provides a fingerprint, leading to missing key observations despite potential detection by other network participants.

While Reassembler supports the first two possibilities, this paper focuses on the second variant, where the attack is automatically inferred from a set of attack fingerprints. The core principle is to use a relatively high detection threshold while leveraging the fact that a node under attack also submits fingerprints. Thus, an attack can be identified by searching for fingerprints that have recorded an attack on themselves while having a high enough detection threshold. In summary, the attack identification step of the Reassembler produces the three outputs parameters summarized in Table II. Both intermediate nodes and

attack sources are not final but serve as a basis for further analysis.

*3) **Attack Analysis**:* An attack fingerprint indicates the absolute number of packets to a target and the detection threshold used to capture the fingerprint. Detection threshold aids in putting the number of packets into context, depending on the AS capacity. For example, a router from a local ISP might have a conspicuously high detection threshold of over 75%. However, if the total attack percentage reveals that this node accounts for less than 0.5%, it becomes less interesting again. Conversely, nodes with a low detection threshold might still be interesting if they account for a large percentage of the total attack. Examples include large DNS servers that can be exploited for a reflection attack.

Distance of attacker. The distance of an attacker to the attack target is calculated based on the TTL values recorded at the target. For the calculation, the knowledge of the initial TTL value of an attack packet is required. Borrowing the idea employed in Hop-Count Filtering [20], the initial TTL value of the packet is derived by finding the smallest common initial TTL value that is higher than the TTL value recorded at the target. This is possible because commonly observed initial TTL values are relatively far apart, but paths between two internet hosts rarely exceed 30 hops in the distance [4], [19]. We consider $32, 64, 128$, and 255 as common initial TTL values, and both the Fingerprint Generator and the Reassembler employ this list for generation and calculation.

Distance of an intermediate node. Compared to calculating the attacker's distance, determining the distance between intermediate nodes and the target does not require knowing the initial TTL value. In this context, calculating the distance is more straightforward, as it simply involves finding the relative difference between observed TTL values for a packet sent to the same target. While the calculation is trivial, some factors can make this calculation imprecise. A network packet is not always guaranteed to take the same route in the real world. This means the TTL values at intermediate nodes and the target may fluctuate slightly. That factor is neglected as the simulated network scenario returns a stable shortest path between any two nodes.

Detecting Spoofed Sources. Utilizing the TTL field of an IP packet for identifying spoofed addresses has proven effective in learning spoofed sources, as shown by Wang *et al.* [20]. The learning phase can rely on the fingerprint aggregation process with the extended fingerprint format. Rather than comparing TTL values with previously recorded values in a database, the detection can be performed solely by examining the fingerprint data. If a fingerprint contains more than one TTL value per source IP, the IP address has likely been spoofed.

*4) **Global Fingerprint**:* The aggregation to build a global fingerprint is the fourth and last step in the Reassembler process. Reassembler provides a compact fingerprint that summarizes the major properties of the attack from a global view. Table III shows the attributes into four categories.

TABLE III: Reassembler's Global Fingerprint Format

Category	Attribute	Data Type
Attack	Start Time	Timestamp
	End Time	Timestamp
	Duration (s)	Float
	Attack Service	String
	Attack Protocol	String
Target	IP Address	String
	Detection Threshold	Float
Intermediate Nodes	Nr. Nodes	Integer
	Discarded Nodes	Integer
	Detection Threshold Percentiles	Map<Integer, Float>
	Key Nodes	Object
Sources	Nr. Sources	Integer
	Pct. Spoofed	Float

Attack properties offer insights about the attack, with attributes such as 'Attack Service' and 'Attack Protocol.' Start and end times of the attack are based on a global perspective, considering only filtered intermediate nodes. The *Target* properties identify the attack target and its configuration parameters. *Intermediate Nodes* form the most significant portion of the global fingerprint. These properties describe the filtered count of intermediate nodes observing the attack traffic and give data on the detection threshold distribution among these nodes. Lastly, *Sources* properties detail the attack's origins, including number of observed IPs targeting during the interval and an estimate of *Percentage of Spoofed Addresses*, assessing whether the reported number of sources is likely inflated or accurate.

IV. Experimental Evaluation

The evaluation of the Reassembler module is based on different networks and adversarial scenarios. From a network topology perspective, two scenarios exist (*cf.* Table IV). The small network *N1* consists of 268 network nodes (*cf.* Figure 2 left). The large network *N2* consists of almost 10'000 network nodes grouped in 15 subnets (*cf.* Figure 2 right).

TABLE IV: Topology Configuration for the Evaluation

Network	# Subnets	Max Levels	Max Clients	# Nodes
N1	5	3	5	268
N2	15	6	5	9'356

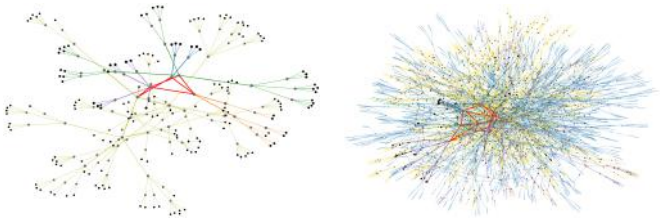

Fig. 2: Visualization of the N1 (left) and N2 (right) networks

Based on the two networks, four different attack scenarios are created with increasing attack size (*cf.* Table V). For each network, a small and a large attack are simulated. With the percentage of spoofed IPs constant at 25%, the

TABLE V: Evaluation Scenarios for N1 and N2

ID	Netw.	#Sources	#Background	Spoofed	Fingerprints
S1	N1	5	10	25%	85
S2	N1	20	50	25%	343
S3	N2	100	200	25%	2'777
S4	N2	500	1000	25%	12'712

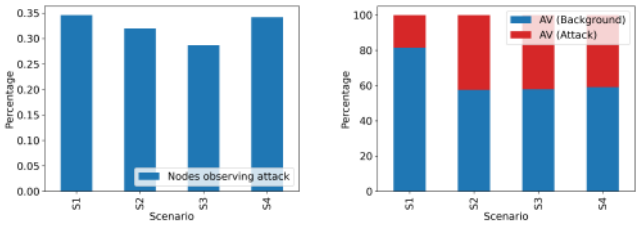

Fig. 3: Left: Attack Vector (AV) Composition. Right: Percentage of Observing Nodes.

number of resulting fingerprints mainly depends on the number of sources and background routes. For the large scenarios (S2, S4), it can be observed that the number of fingerprints generated exceeds the number of nodes in the respective network. For example, scenario S4 generates 12'712 fingerprints in a network (N2) of 9'356 nodes. This means that some of the network nodes submit more than one attack fingerprint.

A. Ground Truth Data

Ground truth data provides insights into how the configuration parameters and the proposed simulated network topology influence generated fingerprints. With the background traffic configuration of the scenarios (*cf.* Table V), it is clear that not all nodes observe actual attack traffic. Figure 3 (right) shows the percentage of nodes observing the attack out of all nodes submitting at least one fingerprint. The percentage values range from 28% to 35%. This aligns with the expected results from the attack configuration, where the number of attack sources is about half of the background traffic for all scenarios. The observed variations can be explained by the randomness in the scenario, where an attack traffic path might have a different length (*i.e.*, involves fewer nodes) than a background traffic path.

Similar fluctuations can also be observed when analyzing the composition of observed attack vectors (*i.e.*, flattened fingerprints). Figure 3 (left) shows a significantly lower percentage of true attack vectors in Scenario 1. This is because three of the five randomly assigned attack sources are located in the same subnet as the attack target, thus producing much shorter attack traffic paths that involve fewer nodes. In the other scenarios, the true and background attack vectors are roughly split into 40% and 60%. The deviation from the expected value of 33% can be attributed to the difference in path length between attack and background traffic paths. Attack sources are only selected from leaf nodes in the subnet tree, but background traffic paths are randomly sampled from the whole network.

B. Reassembler Evaluation

The Reassembler module derives the ground truth data and the attack configuration from fingerprints. This estimates intermediate nodes observing an attack (*cf.* Figure 4 left). Even though the absolute values increase from scenario 1 to scenario 4, the relative number of discarded nodes lies between 19% and 28% for all scenarios.

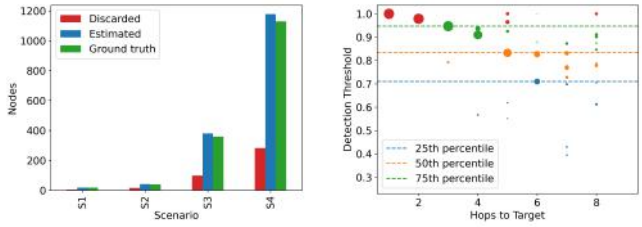

Fig. 4: Left: Intermediate Nodes Estimation. Right: Detection thresholds for Scenario S2.

Another insight derived by the Reassembler is the global distribution of detection thresholds of the participating nodes. A scatter plot of the detection thresholds from the nodes in S2 is shown in Figure 4 (right). The x-axis represents the distance in hops to the attack target. The size of each marker signifies the total count of packets observed by a node, helping in determining whether it is a high-volume node or a comparatively smaller one.

It stands out that up to the fifth hop on the x-axis, one single node per hop is always significantly larger than the others. This aligns with the hierarchical network topology, where, within a subnet, only one path from the root node to a leaf node (*i.e.*, the attack target) exists. At a single hop distance to the target, the detection threshold corresponds to 1 (*cf.* Figure 4), indicating that every packet observed is considered part of the attack. Overall, the detection thresholds shown in Figure 4 are rather high, with only a few outliers below 50%. This is beneficial because a standard DDoS Dissector instance would detect the attack and generate an attack fingerprint.

1) ***IP Spoofing****:* Any strategy that hinders the attack's detection or analysis process is considered adversarial. The attack scenario consists of 100 randomly chosen attack sources within topology N2. Across the different runs, only the spoofed percentage is updated, while the target nodes and the network stay the same. Due to the randomness in setting nodes that use a spoofed IP, an attack source may use spoofed IPs in one run but a real IP in the next. While the attack sources stay the same, the set of nodes that use a spoofed IP is randomly drawn on each run.

The results of the experiment are shown in Figure 5. Within each plot, the spoofed percentage is increased in 5% steps from 0% to 100% using the same network and spoofed IP pool size. Each subplot uses a different IP

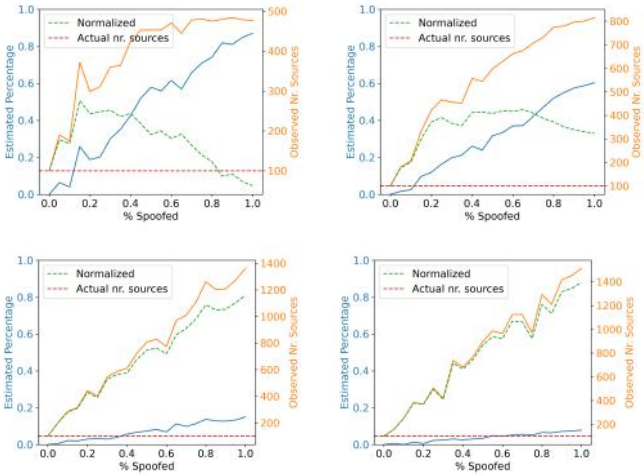

Fig. 5: Spoofed Sources in Scenario S2 with Different Spoofed IP Pool Sizes: Top: 500 (left), 1000 (right). Bottom: 5000 (left), 10000 (right).

Pool Size, ranging from 500 to 10'000. The red horizontal dashed line represents the actual number of sources, which remains consistent across all experimental runs. The orange line denotes the number of observed sources at the attack target. It includes all observed IPs, including spoofed IPs that, for example, have only been observed once. The blue line signifies the estimated percentage of spoofed addresses. Finally, the green dashed line denotes the normalized count of observed sources, calculated by subtracting the estimated spoofed percentage (blue line) from the observed count (orange line).

It can be observed that the number of observed sources rapidly grows with an increasing percentage of spoofed IPs. However, as the estimated percentage (blue line) practically increases linearly with the spoofed percentage on the x-axis, the normalized count (green) settles around 2.5 times the actual count. More than that, the normalized count falls below the actual count for spoofed percentages above 80%. Such behavior is only observed in 500 (top left), which has the smallest spoofed IP pool size.

2) ***Missing Fingerprints****:* With perfect attack coverage, it is possible to see from what distances most of the attack traffic emerged. However, such a scenario is unlikely. Even minor deviations of just one hop in the distance can result in an attack coverage chart that is not logical for certain distances. An example of this is shown in Figure 6 (right), where the observed coverage 7 hops away from the target is over 100% (indicated by the red bar). Naturally, a value over 100% does not make sense and indicates that some distance estimations are incorrect. As a consequence, some nodes are attributed to the wrong group (*i.e.*, distance), which distorts the result.

The experimental setup is based on the configuration of scenario S3 (*cf.* Table V). The key question is how the estimated number of intermediate nodes is affected by

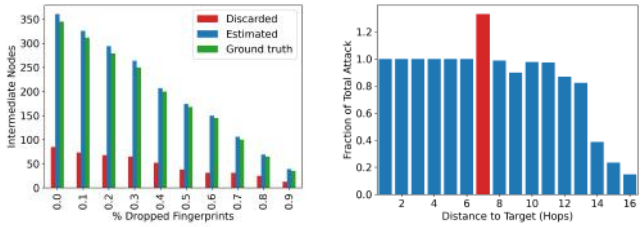

Fig. 6: Left: Intermediate Nodes Estimation Increasing Dropped Fingerprints. Right: Attack Coverage in S4.

missing fingerprints. This is evaluated by comparing the estimated number of intermediate nodes with the ground truth data across multiple runs with an increasing number of dropped fingerprints (*cf.* Figure 6, left).

The number of estimated nodes decreases linearly as the number of excluded fingerprints increases. This trend is positive because excluding specific fingerprints causes no unexpected outliers. In that regard, no minimum amount of participating network nodes is required to make a global analysis work.

3) ***Detection Thresholds****:* The fraction of the total attack decreases when moving away from the attack target. The observation is confirmed by a scatter plot of the observed fractions of the total attack by distance to the attack target (*cf.* Figure 7, left). A linear regression shows that the fraction of the total attack behaves inversely proportional to the distance from the attack target.

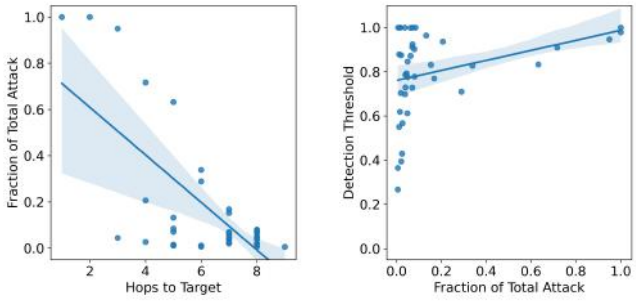

Fig. 7: Left: Fraction of total attack vs. hops to target for scenario S2. Right: Detection threshold vs. fraction of total attack for scenario S2.

Furthermore, the detection threshold is also related to the observed fraction of the attack. As Figure 7 (right) shows, the fraction of total attack behaves proportionally to the detection threshold in Scenario S2. It confirms the intuition that nodes located further away from the attack target perceive a reduced proportion of the total attack and require a lower detection threshold to identify the attack.

To evaluate the impact of background traffic on the required detection thresholds, several runs of S3 are performed, progressively increasing the background traffic. The four runs with 500, 2'000, 5'000, and 10'000 unique background traffic routes are depicted in Figure 8. Each

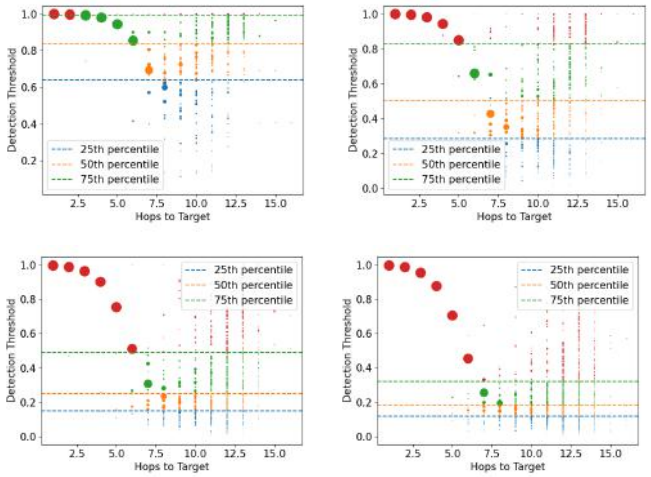

Fig. 8: Detection Thresholds for Scenario S3 Increasing Background Traffic Routes: Top: 500 (left), 2000 (right). Bottom: 5000 (left), 100000 (right).

marker's size represents the network node's size regarding observed packets.

With 500 background routes, 2.5 times the number of attack routes, the median detection threshold is slightly over 80%. The 75th percentile is nearly at a 100% detection threshold, suggesting that many nodes observed only the attack and no background traffic. With 10 times the number of background traffic compared to attack traffic, the median detection threshold decreases to around 50%. In comparison, the 75th percentile with 2'000 background routes remains relatively high at over 80%. Only when increasing the number of background routes to 5'000, all three percentiles for the detection threshold fall below 50%. For 10'000 background routes, which is 50 times the number of attack routes, the mean detection threshold falls below 20%.

V. Summary and Future Work

This paper introduced Reassembler [2], [3], an open-source tool to facilitate global DDoS attack analysis using attack fingerprints. Reassembler is a novel approach that, through a data-driven approach, provides a global, holistic view of DDoS attacks by using and aggregating attack fingerprints from various locations. By extending prior work on DDoS fingerprinting, Reassembler provides a robust foundation for multi-perspective DDoS analysis extracting novel properties, such as the count of intermediate nodes and the estimated percentage of spoofed IPs.

Also, the open-source repositories [2], [3] include all evaluation scripts, enabling additional experiments and further research. Future work will address realistic traffic mix modeling and the inclusion of various attack types to augment the fingerprint generator's efficacy.

References

[1] A. Akella, A. Bharambe, M. Reiter, and S. Seshan, "Detecting DDoS Attacks on ISP Networks," in *Proceedings of the Workshop on Management and Processing of Data Streams*, 2003, pp. 1–2.

[2] J. Brunner, "Reassembler." [Online]. Available: https://github.com/j0nezz/reassembler

[3] ——, "Reassembler Fingerprints: DDoS Dissector." [Online]. Available: https://github.com/j0nezz/ddos_dissector

[4] B. Cheswick, H. Burch, and S. Branigan, "Mapping and Visualizing the Internet." in *USENIX Annual Technical Conference, General Track*. Citeseer, 2000, pp. 1–12.

[5] J. G. Conrads, "DDoS Attack Fingerprint Extraction Tool: Making a Flow-based Approach as Precise as a Packet-based," Master's thesis, University of Twente, 2019.

[6] DDoS Clearing House, "DDoS Dissector." [Online]. Available: https://t.ly/glGxM

[7] ——, "DDoSDB." [Online]. Available: https://t.ly/IHka

[8] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Fingerprinting Internet DNS Amplification DDoS Activities," in *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2014, pp. 1–5.

[9] A. Gruhler, B. Rodrigues, and B. Stiller, "A Reputation Scheme for a Blockchain-based Network Cooperative Defense," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2019)*, April 2019, pp. pp. 71–79, washington, United States of America (USA).

[10] K. Hove, "Automated DDoS Attack Fingerprinting by Mimicking the Actions of a Network Operator," B.S. thesis, University of Twente, 2019.

[11] B. Krebs, "Powerful New DDoS Method Adds Extortion," 2018. [Online]. Available: https://t.ly/6pPP

[12] F.-Y. Lee and S. Shieh, "Defending Against Spoofed DDoS Attacks with Path Fingerprint," *Computers & Security*, Vol. 24, No. 7, pp. 571–586, 2005.

[13] S. Mannhart, B. Rodrigues, E. Scheid, S. S. Kanhere, and B. Stiller, "Toward Mitigation-as-a-Service in Cooperative Network Defenses," in *2018 IEEE 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (CyberSciTech 2018)*, Aug 2018, pp. pp. 362–367, athens, Greece.

[14] R. Mitchell and I.-R. Chen, "A Survey of Intrusion Detection Techniques for Cyber-Physical Systems," *ACM Computing Surveys (CSUR)*, Vol. 46, No. 4, pp. 1–29, 2014.

[15] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-Of-Service Activity," *ACM Transactions on Computer Systems (TOCS)*, Vol. 24, No. 2, pp. 115–139, 2006.

[16] O. A. Osanaiye, "Short Paper: IP Spoofing Detection for Preventing DDoS attack in Cloud Computing," in *2015 18th International conference on intelligence in next generation networks*. IEEE, 2015, pp. 139–141.

[17] B. Rodrigues, E. J. Scheid, C. Killer, M. Franco, and B. Stiller, "Blockchain Signaling System (BloSS): Cooperative Signaling of Distributed Denial-of-Service Attacks," *Journal of Network and Systems Management*, Vol. 28, No. 3, pp. 1–27, August 2020. [Online]. Available: https://doi.org/10.1007/s10922-020-09559-4

[18] B. Rodrigues and B. Stiller, "Cooperative Signaling of DDoS Attacks in a Blockchain-based Network," in *The ACM SIGCOMM 2019 Conference Posters and Demos*. Beijing, China: ACM, August 2019, pp. 39–41. [Online]. Available: https://dl.acm.org/citation.cfm?id=3342300

[19] Y. Vanaubel, J.-J. Pansiot, P. Mérindol, and B. Donnet, "Network Fingerprinting: TTL-Based Router Signatures," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 369–376.

[20] H. Wang, C. Jin, and K. G. Shin, "Defense Against Spoofed IP Traffic Using Hop-count Filtering," *IEEE/ACM Transactions on Networking*, Vol. 15, No. 1, pp. 40–53, 2007.

[21] O. Yoachimik, "Cloudflare DDoS threat report 2022 Q3," 2022. [Online]. Available: https://t.ly/KzM15

[22] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) flooding attacks," *IEEE communications surveys & tutorials*, Vol. 15, No. 4, pp. 2046–2069, 2013.